
AMMAG
CPC PCW - PC
HORS SERIE N°5
Le plein de programmes pour votre AMSTRAD: des jeux, des utilitaires...
SPECIAL LISTINGS
M 1896 - 5 H- 25,00 F-RD
3791896025006 00055
MARS / AVRIL 1988 - 25 F
Anatol.

HORS-SERIE N°5

MARS 1988

SOMMAIRE

LECTEURS

N'usez plus vos doigts !

Il est 3 heures du matin, vos paupières se ferment toutes seules et vous avez tapé un listing dans lequel se trouvent encore des erreurs de saisie ? Pas d'angoisse ! Nous avons pensé à vous... Tournez vite les pages et allez lire notre bon de commande page 5 , en nous le renvoyant dûment rempli et accompagné de son paiement, vous recevrez tous les listings parus dans ce numéro.

Ce "Hors-Série" d'AM-MAG est édité par Laser Presse SARL, 5-7, rue de l'Amiral Courbet - 94160 Saint-Mandé.
Directeur de la publication : Jean Kaminsky.
REDACTION. Directeur de la rédaction : Jacques Eltabet.
Rédacteur en chef : Georges Brize. Secrétaire de rédaction : Suan Ajirent.
Responsable listings : Jean-Claude Paulin. Couverture : Anatol. Illustrations : Thiriet, Mokeït.
FABRICATION. Directeur de la fabrication : Jean-Jacques Galmiche. Secrétaire générale de la rédaction : Françoise Kergreis. Maquettiste : Michel Lhopitault. Montage : Jean-Baptiste Ballériaud.
ADMINISTRATION. Diffusion : Bertrand Desroche. Abonnements : Martine Lapierre au 16 (1) 43.98.01.71. Comptabilité : Sylvie Kaminsky. Assistante de direction : Annette Carniaux.
Régie publicitaire : NEO-MEDIA, 5-7, rue de l'Amiral Courbet, 94160 Saint-Mandé - Tél. : 43.98.22.22.
Chef de publicité : Christine Gourmelon-Malherbe. Assistante de publicité : Mick Deret.
Commission paritaire : en cours. Dépôt légal : 1er trimestre 1988.
Photocomposition : Compo Imprim, 94250 Gentilly. Impression : RBI, Tima Roto.
AM-MAG est une publication strictement indépendante de la société Amstrad.

UTILITAIRE

VÉRIFICATEUR V.2

Vous avez certainement remarqué la présence, dans nos listings, d'un numéro entre crochets ([000]). Ce numéro va vous permettre de pouvoir relire très facilement vos listings et de détecter à coup sûr une éventuelle erreur de saisie. Vous pourrez trouver ci-dessous trois courts programmes correspondant chacun à une version de CPC. Vous devrez donc saisir celui qui vous concerne (sans erreur !) et le sauvegarder. Cet utilitaire vous servira désormais pour tous nos listings, de ce numéro et des suivants. Cette méthode est testée depuis plusieurs mois par notre confrère allemand CPC Schneider International, que nous remercions pour sa collaboration.

Mode d'emploi

Après avoir saisir le programme, faites un "RUN". L'écran doit normalement s'effacer et afficher « VÉRIFICATEUR V.2 installé ». Pour vérifier vos listings, il suffit alors de les charger : |CHECK,2 pour lister le programme avec son numéro de contrôle. Si celui-ci est semblable au nombre entre crochets paru dans nos listings, la ligne est correcte. Si ce numéro diffère (voir exemple) vous avez obligatoirement fait une erreur de saisie... Simple, non ?

De la même manière, vous pourrez tirer sur imprimante votre listing avec tous les numéros de contrôle (comme dans nos listings) en faisant : |CHECK,8. Nous espérons que cette méthode de vérification vous permettra de gagner du temps en relecture (seuls désormais les numéros de contrôle sont à vérifier par rapport à ceux qui sont imprimés dans la revue) et que cet « auto-débbugage » décongestionnera les permanences effectuées par notre technicien. Au travail...

Nota : le signe « | » (barre verticale) s'obtient en appuyant simultanément sur <SHIFT> et <@>.

Exemple

```
10 DATA 2,3,4     [209]
15 REM si erreur
(virgule remplacee par
un point virgule)
10 DATA 2;3,4     [187]
```

```
100 REM Verificateur 464
110 MEMORY &A4FF
120 FOR a%=&A500 TO &A607
130 READ byte$
140 POKE a%,VAL("&"+byte$)
150 NEXT
160 CLS:PRINT
170 PRINT"Verificateur V.2 464
    installe"
180 CALL &A500:|ON
190 :
200 DATA 21,09,a5,01,0d,a5,c3,d1
210 DATA bc,00,00,00,00,18,a5,c3
220 DATA 2a,a5,c3,2f,a5,c3,43,a5
230 DATA 4f,ce,4f,46,c6,43,48,45
240 DATA 43,cb,00,00,cf,98,aa,c3
250 DATA a8,a5,21,27,a5,18,03,21
260 DATA 24,a5,28,06,cd,00,b9,c3
270 DATA 06,dd,11,3a,bd,01,03,00
280 DATA ed,b0,c9,4f,cd,00,b9,0d
290 DATA 28,08,0d,20,ea,dd,7e,02
300 DATA 18,04,7b,11,01,00,cd,a2
310 DATA c1,cd,a3,e7,e5,4e,23,46
320 DATA 23,5e,23,56,e1,78,b1,c8
330 DATA cd,3c,c4,e5,09,e3,cd,63
340 DATA e1,21,a4,ac,cd,7a,a5,e1
350 DATA 18,e2,e5,cd,ba,a5,e3,cd
360 DATA 98,a5,cd,96,f2,e3,cd,f6
370 DATA a5,cd,4e,c3,e1,7e,a7,c8
380 DATA cd,98,a5,cd,4e,c3,18,f5
390 DATA 3a,24,ac,d6,08,47,7e,a7
400 DATA c8,cd,45,e1,23,10,f7,c9
410 DATA cd,24,a5,f5,c5,d5,e5,cd
420 DATA ba,a5,cd,f6,a5,e1,d1,c1
430 DATA f1,c9,eb,1b,af,47,67,6f
440 DATA 2f,32,23,a5,13,1a,d6,30
450 DATA 38,04,fe,0a,38,f6,1a,13
460 DATA a7,c8,4f,3a,23,a5,a1,fe
470 DATA 20,28,f3,79,fe,22,20,07
480 DATA 3a,23,a5,2f,32,23,a5,3a
490 DATA 23,a5,a7,79,c4,8a,ff,4f
500 DATA ad,07,6f,09,18,d8,3e,20
510 DATA cd,5c,c3,3e,5b,cd,5c,c3
520 DATA cd,79,ee,3e,5d,c3,5c,c3
```

```
100 REM Verificateur 664
110 MEMORY &A4FF
120 FOR a%=&A500 TO &A607
130 READ byte$
140 POKE a%,VAL("&"+byte$)
150 NEXT
160 CLS:PRINT
170 PRINT"Verificateur V.2 664
    installe"
180 CALL &A500:|ON
190 :
200 DATA 21,09,a5,01,0d,a5,c3,d1
210 DATA bc,00,00,00,00,18,a5,c3
220 DATA 2a,a5,c3,2f,a5,c3,43,a5
230 DATA 4f,ce,4f,46,c6,43,48,45
240 DATA 43,cb,00,00,cf,02,ac,c3
250 DATA a8,a5,21,27,a5,18,03,21
260 DATA 24,a5,28,06,cd,00,b9,c3
270 DATA 4c,cb,11,5b,bd,01,03,00
280 DATA ed,b0,c9,4f,cd,00,b9,0d
290 DATA 28,08,0d,20,ea,dd,7e,02
300 DATA 18,04,7b,11,01,00,cd,a9
310 DATA c1,cd,69,e8,e5,4e,23,46
320 DATA 23,5e,23,56,e1,78,b1,c8
330 DATA cd,75,c4,e5,09,e3,cd,59
340 DATA e2,21,8a,ac,cd,7a,a5,e1
350 DATA 18,e2,e5,cd,ba,a5,e3,cd
360 DATA 98,a5,cd,58,f3,e3,cd,f6
370 DATA a5,cd,9b,c3,e1,7e,a7,c8
380 DATA cd,98,a5,cd,9b,c3,18,f5
390 DATA 3a,09,ac,d6,08,47,7e,a7
400 DATA c8,cd,22,e2,23,10,f7,c9
410 DATA cd,24,a5,f5,c5,d5,e5,cd
420 DATA ba,a5,cd,f6,a5,e1,d1,c1
430 DATA f1,c9,eb,1b,af,47,67,6f
440 DATA 2f,32,23,a5,13,1a,d6,30
450 DATA 38,04,fe,0a,38,f6,1a,13
460 DATA a7,c8,4f,3a,23,a5,a1,fe
470 DATA 20,28,f3,79,fe,22,20,07
480 DATA 3a,23,a5,2f,32,23,a5,3a
490 DATA 23,a5,a7,79,c4,ab,ff,4f
500 DATA ad,07,6f,09,18,d8,3e,20
510 DATA cd,a3,c3,3e,5b,cd,a3,c3
520 DATA cd,49,ef,3e,5d,c3,a3,c3
```

```
100 REM Verificateur 6128
110 MEMORY &A4FF
120 FOR a%=&A500 TO &A607
130 READ byte$
140 POKE a%,VAL("&"+byte$)
150 NEXT
160 CLS:PRINT
170 PRINT"Verificateur V.2 6128
    installe"
180 CALL &A500:|ON
190 :
200 DATA 21,09,a5,01,0d,a5,c3,d1
210 DATA bc,00,00,00,00,18,a5,c3
220 DATA 2a,a5,c3,2f,a5,c3,43,a5
230 DATA 4f,ce,4f,46,c6,43,48,45
240 DATA 43,cb,00,00,cf,02,ac,c3
250 DATA a8,a5,21,27,a5,18,03,21
260 DATA 24,a5,28,06,cd,00,b9,c3
270 DATA 49,cb,11,5e,bd,01,03,00
280 DATA ed,b0,c9,4f,cd,00,b9,0d
290 DATA 28,08,0d,20,ea,dd,7e,02
300 DATA 18,04,7b,11,01,00,cd,a6
310 DATA c1,cd,64,e8,e5,4e,23,46
320 DATA 23,5e,23,56,e1,78,b1,c8
330 DATA cd,72,c4,e5,09,e3,cd,54
340 DATA e2,21,8a,ac,cd,7a,a5,e1
350 DATA 18,e2,e5,cd,ba,a5,e3,cd
360 DATA 98,a5,cd,53,f3,e3,cd,f6
370 DATA a5,cd,98,c3,e1,7e,a7,c8
380 DATA cd,98,a5,cd,98,c3,18,f5
390 DATA 3a,09,ac,d6,08,47,7e,a7
400 DATA c8,cd,1d,e2,23,10,f7,c9
410 DATA cd,24,a5,f5,c5,d5,e5,cd
420 DATA ba,a5,cd,f6,a5,e1,d1,c1
430 DATA f1,c9,eb,1b,af,47,67,6f
440 DATA 2f,32,23,a5,13,1a,d6,30
450 DATA 38,04,fe,0a,38,f6,1a,13
460 DATA a7,c8,4f,3a,23,a5,a1,fe
470 DATA 20,28,f3,79,fe,22,20,07
480 DATA 3a,23,a5,2f,32,23,a5,3a
490 DATA 23,a5,a7,79,c4,ab,ff,4f
500 DATA ad,07,6f,09,18,d8,3e,20
510 DATA cd,a0,c3,3e,5b,cd,a0,c3
520 DATA cd,44,ef,3e,5d,c3,a0,c3
```

AMSAISIE

Nous vous félicitons d'être de plus en plus nombreux à nous communiquer des programmes réalisés partiellement ou totalement en langage machine. Les avantages de ce type de programmation ne sont plus à démontrer, ainsi que l'exceptionnelle qualité de la plupart des créations. Malheureusement, la saisie par le lecteur d'une foule considérable de codes machines n'est pas une partie de plaisir. AMSAISIE a donc été créé dans le but louable de vous faire profiter de quelques petites merveilles, en vous évitant un maximum de contraintes.

Il doit être conservé précieusement et utilisé chaque fois que vous découvrirez dans nos colonnes, un listing de codes machines au format approprié. Exemple :

Adresse :	Codes	Somme de contrôle
8 FAO :	00 00 00 00 00 00 00 00	2F



Mode d'emploi

Après lancement, spécifiez en hexadécimal (sans le préfixe '&') l'adresse de début d'implantation du L.M. Celle-ci s'affiche, suivie de ':' et d'un curseur clignotant. Entrez la série de huit codes, sans vous préoccuper des espaces et sans validez par RETURN (tout cela se réalise automatiquement). En fin de ligne et à l'affichage de ':', entrez la somme de contrôle correspondant à la ligne saisie. En l'absence d'erreur, l'adresse suivante s'affiche, etc. Dans le cas contraire, un signal sonore retentit et le message 'ERREUR !' vous signale une bévue. Pas de panique, puisque l'adresse de la ligne litigieuse se réaffiche. Il suffit donc de retaper cette dernière. De plus, sachez qu'en cours de saisie, la touche DEL est opérationnelle.

Maintenant, comment sauvegarder sur disquette (ou cassette) le langage machine aussi facilement entré en mémoire ? Tout simplement par la commande 'S', qui vous réclamera préalablement le nom du programme à sauver. Toutefois, deux solutions s'offrent à vous.

— Vous êtes fou : et venez de saisir en une seule fois, la 'totalité' des codes hexadécimaux (ouf !). Pas de problème, appuyez sur 'S' et votre programme se trouvera entièrement sauvegardé, après l'entrée du nom de fichier et l'appui sur <RETURN>.

— Vous êtes raisonnable : et désirez morceler votre saisie. Au moment de stoper momentanément votre ouvrage pour le poursuivre ultérieurement, appuyez sur 'S' et attribuez un numéro d'ordre à votre nom de

fichier (exemple : PENDU1). A la fin de la sauvegarde, l'adresse suivante s'affiche ; notez-la. Cette adresse sera 'l'adresse de début' qu'il conviendra de spécifier lors de la reprise de votre travail (PENDU2). Créez de la sorte, une suite de fichiers binaires (PENDU1, PENDU2, PENDU3, etc.). Finalement, chargez à la suite tous les fichiers (n'oubliez pas le MEMORY adéquat) et effectuez une sauvegarde totale et définitive, par la commande indiquée dans le mode d'emploi du programme (exemple : SAVE"PENDU.BIN",b,&8DDO, &10AC. Enfantin !

```
10 ' * AMSAISIE par Denis JARRIL *      [1660]
20 MEMORY &2000:DIM O$(18):MODE 1:B [17671]
ORDER 0:INK 0,0:INK 1,13:CLS:PRINT:
PRINT" I pour changer l'adresse cou
rante":PRINT" S pour sauver les don
nees":PRINT" Tapez les caracteres s
ans espace ni      return (tout se f
ait automatiquement).":PRINT
30 PRINT:INPUT" ADRESSE DE DEPART : [4110]
 ",A$:D$=A$:IF A$="" THEN 30
40 A=VAL("&"+A$)                     [1273]
50 I=0:A$=HEX$(A,4):PRINT:PRINT A$; [3200]
":";:C=VAL("&"+LEFT$(A$,2))+VAL("&"
+RIGHT$(A$,2))
60 T$="":WHILE T$="":CALL &BB8A:T$= [3454]
INKEY$:CALL &BB8D:WEND:T$=UPPER$(T$
)
70 IF T$="I" THEN CLS:RUN           [1132]
80 IF T$<>"S" THEN 110 ELSE D=VAL(" [3318]
&"+D$):IF D>0 AND A<0 THEN A=A+6553
6
90 PRINT:PRINT:INPUT"  NOM : ",N$:I [3394]
F N$<>"" THEN SAVE N$,B,D,A-D+1
100 GOTO 50                          [384]
110 IF T$<>CHR$(127) THEN 130 ELSE  [7405]
IF I=0 THEN 60 ELSE I=I-1:PRINT CHR
$(8);" ";CHR$(8);:IF I/2<>ROUND(I/2
) THEN PRINT CHR$(8);" ";CHR$(8);
120 GOTO 60                          [390]
130 IF T$<"0" OR T$>"F" THEN SOUND  [2693]
7,150,20:GOTO 60
140 IF T$<"A" AND T$>"9" THEN SOUND [2632]
 7,150,20:GOTO 60
150 PRINT T$;:IF I=15 THEN PRINT":" [3278]
; ELSE IF I/2<>ROUND(I/2) THEN PRIN
T" ";
160 O$(I)=T$                         [592]
170 I=I+1:IF I<18 THEN 60           [1324]
180 FOR I=0 TO 15 STEP 2:X=VAL("&"+ [4450]
O$(I)+O$(I+1)):POKE A,X:A=A+1:C=C+X
:NEXT:C=C AND &FF
190 IF C=VAL("&"+O$(I)+O$(I+1)) THE [2426]
N 50
200 SOUND 7,50,10:SOUND 7,500,10:PR [3493]
INT"ERREUR!";:GOTO 40
```



JEU

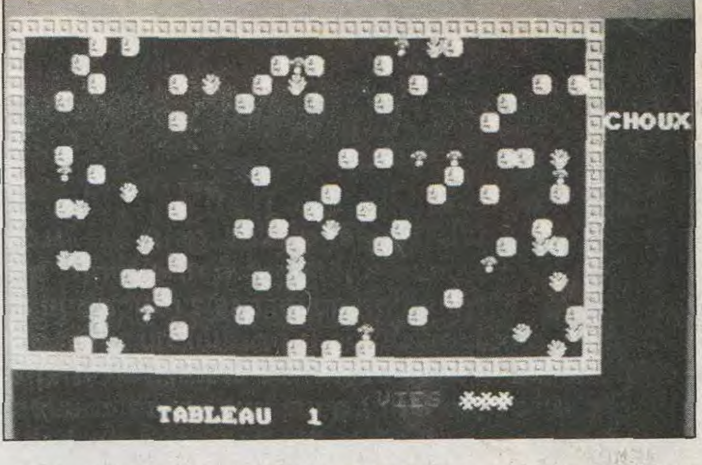
CHENILLE (6128)

éroïne bien connue de la mythologie logicielle, la sempiternelle chenille vorace et dangereusement extensible, reste un des grands plaisirs de notre enfance. Comment résister à l'efficacité d'un aussi court programme, agréable dispensateur d'une joie simple et salutaire.

Benjamin Sentou

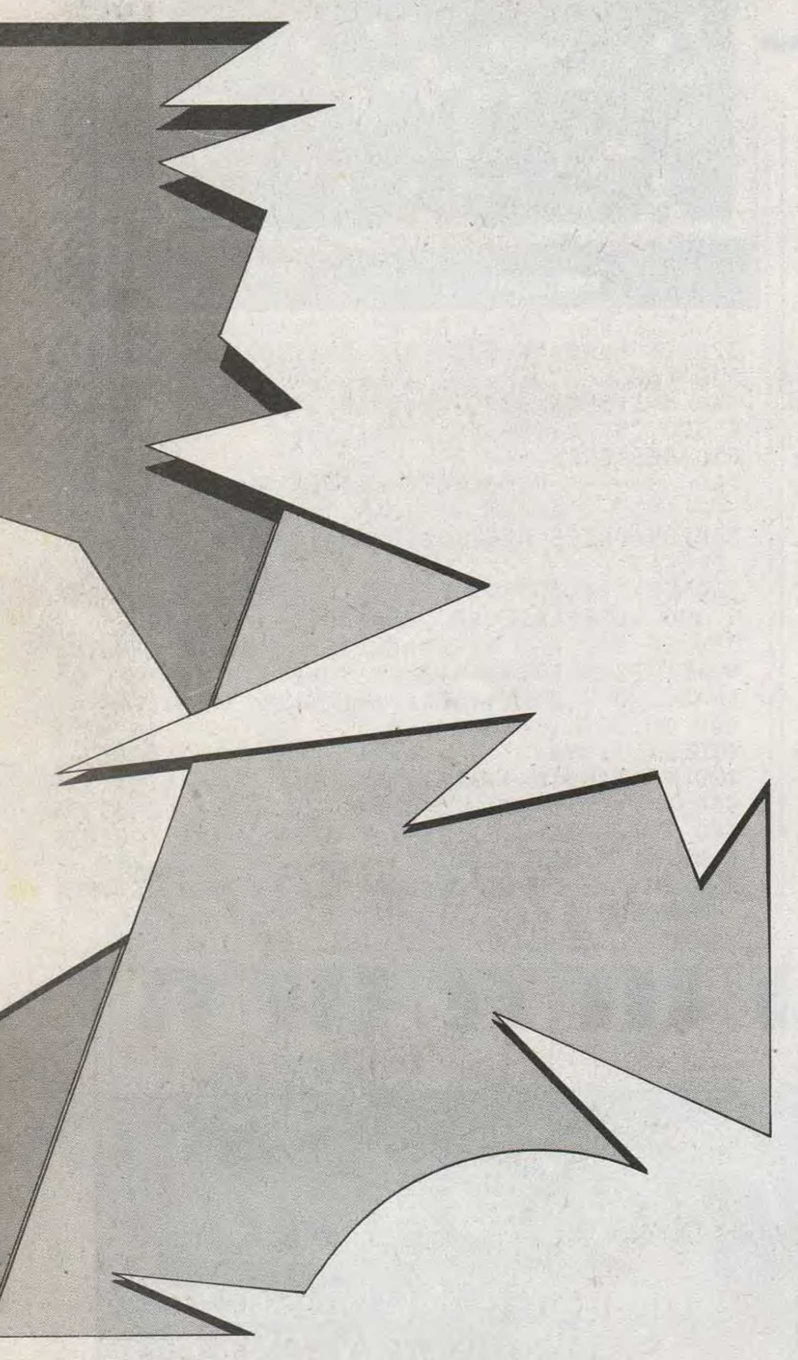
```
10 MODE 1:INK 0,0:BORDER 2:INK 1,6: [3247]
INK 2,21:INK 3,11
20 GOSUB 880                         [1060]
30 DIM X(1000),Y(1000)               [725]
40 SYMBOL AFTER 127                  [1453]
50 SYMBOL 128,36,24,153,102,102,153 [2454]
,24,36
60 SYMBOL 129,28,2,17,33,33,17,2,28 [2165]
70 SYMBOL 130,255,129,253,221,221,1 [1968]
97,253,255
80 SYMBOL 131,126,215,209,159,249,1 [2309]
39,235,126
90 SYMBOL 132,40,42,146,101,58,170, [2456]
84,56
100 SYMBOL 133,0,8,14,0,6,4,4        [977]
110 SYMBOL 134,60,106,189,247,24,36 [2444]
,36,24
120 SYMBOL 135,8,20,66,8,0,24,24    [1740]
130 SYMBOL 136,8,73,42,133,100,34,7 [2151]
4,73
140 PEN 1:LOCATE 23,22:PRINT "VIES" [1435]
150 NM=70:NC=15:ND=9:TA=1:VIE=3     [1839]
160 PEN 3:FOR I=1 TO 35:LOCATE I,1: [4296]
PRINT CHR$(130):LOCATE I,20:PRINT C
HR$(130):NEXT
170 FOR I=1 TO 20:LOCATE 1,I:PRINT  [3760]
```

```
CHR$(130):LOCATE 35,I:PRINT CHR$(13
0):NEXT
180 FOR I=1 TO NM:X=INT(RND*33)+2:Y [4194]
=INT(RND*18)+2:LOCATE X,Y:PEN 2:PRI
NT CHR$(133):LOCATE X,Y:PEN 3,1:PRI
NT CHR$(131):NEXT
190 FOR I=1 TO ND:X=INT(RND*33)+2:Y [7948]
=INT(RND*18)+2:LOCATE X,Y:PEN 0,0:P
RINT " ":LOCATE X,Y:PEN 2:PRINT CHR
$(135):LOCATE X,Y:PEN 1,1:PRINT CHR
$(134):NEXT
200 PEN 2,0:FOR I=1 TO NC*1.2       [847]
210 LOCATE INT(RND*33)+2,INT(RND*18 [2459]
)+2:PRINT CHR$(132)
220 NEXT                            [350]
230 PEN 2:LOCATE 36,6:PRINT "CHOUX" [1681]
240 LOCATE 10,23:PEN 3:PRINT "TABLE [1657]
AU ";TA
250 CH=NC                           [769]
260 X=320:Y=206:L=2:I=1             [1721]
270 X(1)=320:Y(1)=206               [1354]
280 PEN 2:LOCATE 28,22:FOR B=1 TO V [4320]
IE:PRINT CHR$(128);:NEXT:PRINT " "
290 M=1                             [184]
300 ENV 1,15,-1,1,5,1,1:ENV 2,3,0,1 [5621]
,17,1,1:ENV 3,16,1,1,4,-1,1:ENV 4,8
,0,1,3,-1,1,9,0,1:ENV 5,4,-1,5:ENV
6,4,-1,1,12,0,1,4,1,1
310 RESTORE 870:FOR M=1 TO TA-S:REA [4862]
D FLAG1,FLAG2,H,Q:NEXT:IF H=2 THEN
S=S+8:GOTO 310
```




```
320 IF FLAG1=1 THEN SOUND 17,0,H,0  [1971]
330 TAG                             [318]
340 M=1:MOVE 320,206:PRINT " ";:MOV [3364]
E 320,222:PRINT " ";
350 RESTORE 710                     [733]
360 '-------PROGRAMME PRINCIPAL---- [1343]
---
370 GRAPHICS PEN 1,0:GRAPHICS PAPER [1397]
 0
380 EM=M:E=JOY(0):IF E=6 OR E=5 THE [3516]
N 380 ELSE IF E=16 THEN 560
390 IF E>0 AND E<9 THEN IF E<>M THE [4475]
N IF ABS(M-E)<>1 AND ABS(M-E)<>4 TH
EN SOUND 4,20,20,15,1:M=E
400 ON SQ(1) GOSUB 670              [574]
410 AX=X:AY=Y                       [872]
420 X=X+16*(M=4)-16*(M=8)           [1716]
430 Y=Y+16*(M=2)-16*(M=1)           [1379]
440 T=TEST(X+8,Y):IF T=3 THEN X=AX: [2878]
Y=AY:M=EM:GOTO 380
450 IF T=2 THEN FOR K=50 TO 30 STEP [6077]
 -2:SOUND 4,K,1,15:SOUND 4,K*1.5,1,
```



```
15:NEXT:L=L+3:CH=CH-1:SC=SC+5:MOVE
570,298:PRINT CH;:GOTO 480
460 IF T=1 THEN 560                    [494]
470 MOVE X(I-L+1),Y(I-L+1):PRINT "     [1591]
";
480 I=I+1:X(I)=X:Y(I)=Y                [1170]
490 MOVE AX,AY:PRINT CHR$(129);        [1819]
500 MOVE X,Y:PRINT CHR$(128);          [1869]
510 IF CH=0 THEN 770                   [793]
520 GOTO 380                           [488]
530 READ L1,L2,R:IF L1=0 THEN RESTO    [1645]
RE:GOTO 530
540 SOUND R,L1,5,15,1,1:SOUND R+2,L    [4082]
2,3,15,1,1:RETURN
550 DATA 60,100,66,40,110,2,70,120,    [1879]
2,45,130,2,100,140,2,0,0,0
560 SOUND 129,0:SOUND 130,0:SOUND 1    [2505]
32,0
570 FOR K=26 TO 5 STEP -1:INK 1,K:I    [4018]
NK 0,K-5:SOUND 1,670-K*25,3,15:OUT
256,0:NEXT:INK 1,6,3
580 PEN 1,0                            [588]
590 FOR K=I TO I-L+1 STEP -1           [1076]
600 MOVE X(K),Y(K):PRINT CHR$(136);    [2274]
610 FOR J=15 TO 3 STEP -2:SOUND 17,    [4166]
1000,1,J,0,0,10:SOUND 10,500,1,J,0,
0,30:NEXT
620 MOVE X(K),Y(K):PRINT " ";          [1694]
630 NEXT                               [350]
640 TAGOFF:VIE=VIE-1:IF VIE>-1 THEN    [3634]
 INK 1,6:GOTO 260
650 CLS:PEN 3,0:LOCATE 10,10:PRINT     [3351]
"SCORE :";:PEN 2:PRINT SC;:PEN 1:PR
INT " POINTS"
660 CLEAR INPUT:CALL &BB06:RUN         [1207]
670 READ NO:IF NO=1 THEN RESTORE 71    [2232]
0:GOTO 670
680 IF NO=0 THEN N=0:NO2=0 ELSE N=1    [3901]
:IF FLAG2=1 THEN NO2=NO-2 ELSE NO2=
NO
690 SOUND 17,NO,20,15*N,Q*N:SOUND 1    [3563]
0,NO2,20,0,Q*N
700 RETURN                             [555]
710 DATA 478,0,478,0,478,0,478,0,47    [3622]
8,506,478,426,379,0,379,0,319,338,3
19,239,190,0,190,0
720 DATA 213,190,179,253,213,0,213,    [2703]
0,319,338,319,253,213,0,213,0
730 DATA 179,190,179,127,106,0,106,    [2484]
0,113,106,95,127,119,0,119,0
740 DATA 95,106,95,89,80,0,80,0,71,    [2297]
80,89,95,106,0,106,0
750 DATA 106,113,106,95,89,0,89,0,8    [3036]
0,89,95,106,119,0,119,0
760 DATA 119,127,119,106,95,0,95,0,    [6319]
89,95,106,119,127,0,0,0,142,0,0,0,1
59,0,0,0,179,0,0,0,190,0,0,0,213,0,
0,0,1
770 RESTORE 860:TA=TA+1:NM=NM+10:NC    [5297]
=NC+2:ND=ND+1:SOUND 130,0:SOUND 129
,0
780 TAGOFF:PEN 2,0                     [1406]
790 FOR I=2 TO 19                      [622]
800 FOR K=2 TO 18 STEP 16              [1066]
810 LOCATE K,I:PRINT SPC(17)           [1179]
820 READ NO:IF NO=0 THEN RESTORE 86    [1464]
0:GOTO 820
830 SOUND 17,NO,15,15:SOUND 10,338,    [4650]
15,15:SOUND 17,0,2,0:SOUND 10,0,2,0
840 NEXT K,I                           [376]
850 GOTO 180                           [401]
860 DATA 451,451,451,451,451,451,45    [7126]
1,451,426,426,426,426,426,426,426,4
26,402,402,402,402,402,402,402,402,
426,426,426,426,426,426,426,426,0
870 DATA 0,1,0,1,1,0,30,2,0,1,0,3,1    [3357]
,0,20,1,0,1,0,4,1,0,10,5,0,1,0,6,0,
1,0,5,2,2,2,2
880 PRINT:PRINT " CE JEU NE PEUT SE    [8844]
 JOUER QU'AVEC L'AIDE":PRINT "DU JO
YSTICK.SI VOUS ETES BLOQUE,APPUYEZ
SUR LE BOUTON DU TIR (SUICIDE!)."
890 PRINT:PRINT "VOUS POUVEZ MODIFI    [4143]
ER TOUS LES PARAMETRES EN EDITANT L
A LIGNE 150."
900 LOCATE 13,12:PEN 1:PRINT "BONNE    [4293]
 CHANCE ":CALL &BB06:CLS:RETURN
```

AUTEURS

Nous attendons vos réalisations !

Comment procéder ? Vous êtes auteur d'un programme et vous souhaitez le voir être publié. Il vous suffit de nous envoyer vos créations (originales) sur support cassette ou disquette, accompagnées d'un mode d'emploi du logiciel, d'une liste de variables et du bon, ci-dessous, dûment complété. Toutes les propositions seront examinées.



Tous nos listings parus sont rémunérés !

DEMANDE DE PARUTION

NOM .... PRÉNOM ....

ADRESSE ....

....

TÉLÉPHONE ....

NOM DU PROGRAMME ....

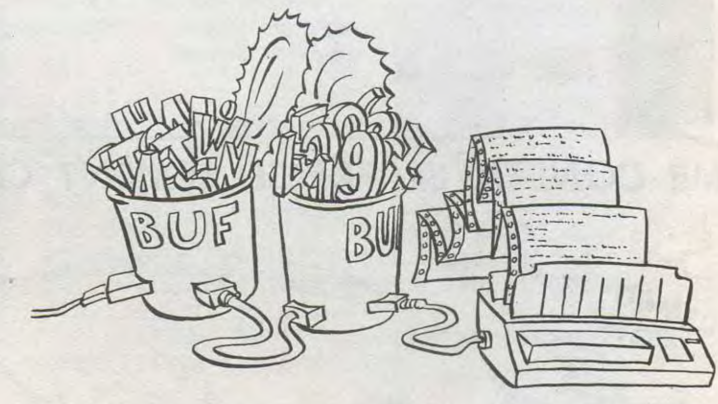
Je certifie sur l'honneur, être l'auteur de ce programme et autorise les éditions AM-MAG à le publier (programme rétribué). Date et signature obligatoires (signature des parents ou d'un tuteur légal pour les mineurs) :

BUFFER IMPRIMANTE

Face à la gabegie temporelle imputable à l'impression, qui n'a pas piaffé d'impatience devant un clavier momentanément inopérant ? Reconnu d'utilité publique et agréé par le ministère de la santé, l'utilitaire de type BUFFER accroît l'efficacité de l'homme moderne, tout en lui évitant les conséquences fatales d'un état nerveux exacerbé.
*Ce programme, gérant le second banc mémoire de 64 Ko du 6128 comme un buffer d'imprimante, permet la poursuite du travail sur ordinateur (programmer, charger ou sauver un fichier, etc.) pendant le « vidage » dudit buffer au rythme de l'impression. Fonctionnant même sous TASWORD, il peut faire gagner un temps précieux dans le cas d'un courrier répétitif. Lors de l'impression, il est possible, par la commande **I BUFFER**, n, de désactiver le buffer et de le réactiver au point où il a été interrompu. **I BCLEAR** efface le contenu du buffer. Les possesseurs de 464-664, disposent ici d'une version adaptée à leur matériel.*

Rémy Jaffer

```
10 '******************************** [1223]
20 '* Buffer-imprimante 32-48-64K * [1229]
30 '** PAR JAFER REMY  (C)1987   ** [1149]
40 '**        Version 6128       ** [740]
50 '**                           ** [104]
60 '* |BCLEAR   : Vide le buffer  * [1212]
70 '* |BUFFER,0 : Desact. le buf. * [1988]
80 '* |BUFFER,1 : React. le buf.  * [2083]
90 '******************************** [1223]
100 :                              [174]
110 MODE 2:MEMORY &3FFF:L=290:M=&30 [1893]
00
120 PRINT"Chargement des DATA en co [3204]
urs":PRINT
130 S=0:READ A$:IF A$="*" THEN 180  [1996]
140 FOR I=1 TO LEN(A$) STEP 2:D=VAL [3099]
("&"+MID$(A$,I,2))
150 POKE M,D:S=S+D:M=M+1:NEXT       [377]
160 READ S$:IF S<>VAL("&"+S$) THEN  [3692]
E=1:PRINT CHR$(7)"ERREUR EN LIGNE :
";L
170 L=L+5:GOTO 130                  [331]
180 IF E=1 THEN STOP                [1650]
190 PRINT"Voulez-vous une sauvegard [4717]
e du code machine"
200 PRINT"(O/N)?";                  [1000]
210 A$=UPPER$(INKEY$):IF A$="" THEN [1810]
 210
220 IF A$="O" THEN SAVE"CODE",B,&30 [1666]
00,736:GOTO 240
230 IF A$<>"N" THEN 210             [871]
240 PRINT A$:PRINT                  [1612]
250 PRINT"Capacite du buffer-imprim [5615]
ante : 1=32K 2=48K 3=64K"
260 INPUT "(1-3)";a:IF a<1 OR a>3 T [2121]
HEN 260
270 CALL &3000,a-1:MEMORY &A57F:NEW [1645]
280 :                              [174]
290 DATA FE01C0DD7E00FE03,41B      [1245]
300 DATA D0F3F53CF5473E03,471      [927]
310 DATA 90C6C4321830F501,38A      [566]
320 DATA C47FED4921A83111,384      [1579]
330 DATA 0040013301EDB021,233      [1485]
```




```
340 DATA E0301180A501C800,30F      [1037]
350 DATA EDB02128BD111B41,310      [1403]
360 DATA 010C00C5E5EDB021,375      [578]
370 DATA 2741D1C1EDB0F132,4BA      [2029]
380 DATA DCA532FAA53209A6,433      [1314]
390 DATA 32BD403217A63C32,28C      [608]
400 DATA 2BA647F13229A680,38A      [1275]
410 DATA 32B540CD07A6F311,3A5      [1325]
420 DATA AFA5219FA5010081,33B      [1157]
430 DATA CDE0BC2195A50180,445      [1125]
440 DATA A5CDD1BCF1872199,531      [829]
450 DATA 30856F7E32B23023,2D9      [1577]
460 DATA 7E32B330219F307E,301      [829]
470 DATA B7283423CD5ABB18,330      [1259]
480 DATA F63332343836340C,23D      [585]
490 DATA 4255464645522D49,230      [574]
500 DATA 4D5052494D414E54,268      [1169]
510 DATA 452030304B0D0A45,16C      [1841]
520 DATA 4E20464F4E435449,231      [1137]
530 DATA 4F4E0D0A0A0021,E9         [1740]
540 DATA EDB0224000EC932,338       [1174]
550 DATA 4200210030110130,D5       [727]
560 DATA 01E80377FBC34000,361      [896]
570 DATA 88A5C3B4A5C3B9A5,56A      [1552]
580 DATA 4255464645D24243,2BF      [1502]
```

ADDITIF TECHNIQUE

Chaque caractère destiné à l'imprimante est dirigé vers une "pending queue", libèrant ainsi l'ordinateur de l'attention du signal "busy" de l'imprimante. Cette file d'attente est vidée à raison de 5 x 300 caractères par seconde à la demande de l'imprimante.

Appels systèmes
&BCD1 : *installation des RSX*
&BB5A : *sortie d'un caractère sur l'écran*
&BD28 : *1er adresse des vecteurs imprimante*
&BCEO : *installe un évènement au 1/300e sec.*
&BCE6 : *enlève un évènement au 1/300e sec.*

Périphériques
&F5xx : *port B du circuit d'entrée/sortie 8255*
&7FCO : *commutation du "banc" standard en &4000*
&7FC4 : *banc 1 en &4000*
&7FC5 : *banc 2 en &4000*
&7FC6 : *banc 3 en &4000*
&7FC7 : *banc 4 en &4000*
Les organigrammes 1 et 2 figurent l'algorithme général utilisé.

Initialisation
Installation des nouveaux vecteurs
Vidage du contenu du buffer
Installation d'un évènement au 1/300 de sec
Installation de nouvelles RSX
Sortir le message indiquant le nbre de Ko utilisés
Retour

Organigramme n° 1

Evènement au 1/300 de sec
Lire un caractère dans le buffer
Buffer vide?
oui
Imprimante prête?
non
Enlever un caractère du buffer et l'envoyer vers l'imprimante
5 caractères sortis?
non
Retour

Organigramme n° 2

```
590 DATA 4C4541D200000000,1A4      [1241]
600 DATA 0033413341C4C400,270      [1435]
610 DATA 0000000000000000,0        [964]
620 DATA 00000000000000CD,CD       [931]
630 DATA D2A50040CDD2A52D,428      [1075]
640 DATA 40CDD2A56040CDD2,4C3      [1516]
650 DATA A56540CDD2A57140,43F      [1422]
660 DATA CDD2A5D740CDD2A5,59F      [1249]
670 DATA EB40F3E3D557D5C5,5C7      [805]
680 DATA 5E235601C47FED49,351      [2002]
690 DATA ED53E5A5CDE4A501,521      [1048]
700 DATA C07FED49C1D17AD1,552      [1304]
710 DATA E1FBC9C5067FED49,525      [790]
720 DATA 1A0EC4ED49C1C9C5,471      [1552]
730 DATA 067FED491218F2F3,3CA      [1140]
740 DATA 01C47FED49213341,30F      [1089]
750 DATA 229BA52299A53EC4,3C4      [1267]
760 DATA 329EA5329DA51134,32E      [1187]
770 DATA 4101CC3E36FFEDB0,41E      [1296]
780 DATA 06030EC4C5067FED,312      [1377]
790 DATA 4921004011014001,FD       [1560]
800 DATA FF3F36FFEDB0C10C,4DD      [387]
810 DATA 10EA01C07FED49C9,439      [427]
820 DATA 0605C53A9DA54FED,388      [1651]
830 DATA 5B99A5CDF3A5FEFF,5FB      [2104]
840 DATA 2819CDF74030143E,2C7      [715]
850 DATA FFCDFFA579CDA740,59D      [1265]
860 DATA 329DA5ED5399A5C1,4B3      [1098]
870 DATA 10D8C9C1C9FE01C0,4FA      [908]
880 DATA DD7E00B728182127,29A      [1247]
890 DATA 411128BD010C00ED,231      [728]
900 DATA B011AFA5219FA501,37B      [1377]
910 DATA 0081CDE0BCC9211B,3EF      [1426]
920 DATA 411128BD010C00ED,231      [728]
930 DATA B0219FA5CDE6BCC9,54D      [1388]
940 DATA B7C0C307A6212741,370      [1128]
950 DATA 1128BD010C00EDB0,2A0      [466]
960 DATA C96F3A9EA54FED5B,44C      [1535]
970 DATA 9BA5CDF3A5FEFFC2,664      [731]
980 DATA A5407D21C2404604,2CF      [1285]
990 DATA 05280A23BE2320F8,253      [1081]
1000 DATA 7EFEFF280ECDFFA5,522     [1853]
1010 DATA 79CDA740329EA5ED,48F     [764]
1020 DATA 539BA537C9A7C94F,452     [1435]
1030 DATA 13AF933E809A79C0,3E6     [1460]
1040 DATA 1133413CFEC82804,2B3     [1104]
```

```
1050 DATA 110040C93EC41133,260          [501]
1060 DATA 41C90AA05EA15CA2,3B1          [1735]
1070 DATA 7BA323A640AB7CAC,3FA          [1028]
1080 DATA 7DAD7EAE5DAF5B3A,3F7          [1477]
1090 DATA 9EA54FED5B9BA5CD,4E7          [2083]
1100 DATA F3A5FEFFC2E940A7,627          [1961]
1110 DATA C937C96FED5B9BA5,4C0          [1526]
1120 DATA 3A9EA54F7D189ECD,3CC          [1627]
1130 DATA 104130023FC9C506,256          [1425]
1140 DATA EFE67FED79F680ED,61D          [1179]
1150 DATA 79E67FED79C137C9,505          [1193]
1160 DATA C54F06F5ED781717,3A2          [1666]
1170 DATA 79C1C90000000000,203          [856]
1180 DATA 00000000000000C3,C3           [859]
1190 DATA BEA5C3C3A5C3C8A5,5BE          [1044]
1200 DATA C3CDA50000000000,235          [1393]
1210 DATA *                             [109]
```

```
10 '******************************** [1223]
20 '* Buffer-imprimante  8-16-24K * [1469]
30 '** PAR JAFER REMY  (C)1987    ** [1149]
40 '**       Version 464-664      ** [553]
50 '**                            ** [104]
60 '* |BCLEAR   : Vide le buffer   * [1212]
70 '* |BUFFER,0 : Desact. le buf.  * [1988]
80 '* |BUFFER,1 : React. le buf.   * [2083]
90 '******************************** [1223]
100 :                               [174]
110 MODE 2:MEMORY &3FFF:L=290:M=&30 [1893]
00
120 PRINT"Chargement des DATA en co [3204]
urs":PRINT
130 S=0:READ A$:IF A$="*" THEN 180  [1996]
140 FOR I=1 TO LEN(A$) STEP 2:D=VAL [3099]
("&"+MID$(A$,I,2))
150 POKE M,D:S=S+D:M=M+1:NEXT       [377]
160 READ S$:IF S<>VAL("&"+S$) THEN  [3692]
E=1:PRINT CHR$(7)"ERREUR EN LIGNE :
";L
170 L=L+5:GOTO 130                  [331]
180 IF E=1 THEN STOP                [1650]
190 PRINT"Voulez-vous une sauvegard [4717]
e du code machine"
200 PRINT"(O/N)?";                  [1000]
210 A$=UPPER$(INKEY$):IF A$="" THEN [1810]
 210
220 IF A$="O" THEN SAVE"CODE",B,&30 [2386]
00,552:GOTO 240
230 IF A$<>"N" THEN 210             [871]
240 PRINT A$:PRINT                  [1612]
250 PRINT"Capacite du buffer-imprim [6010]
ante : 1=8K 2=16K 3=24K"
260 INPUT "(1-3)";a:IF a<1 OR a>3 T [2121]
HEN 260
270 CALL &3000,a-1:MEMORY 42239-A*8 [2054]
192:NEW
280 :                               [174]
```

```
290 DATA FE01C0DD7E00FE03,41B           [1245]
300 DATA D0F3F5473E039021,3F1           [971]
310 DATA 002511002019 3D20,CC           [1226]
320 DATA FCE521B9301100A5,3A1           [820]
330 DATA 016E01EDB02128BD,313           [1507]
340 DATA 1156A6010C00C5E5,2C4           [1020]
350 DATA EDB02162A6D1C1ED,545           [1489]
360 DATA B0E1222DA5CD2FA5,426           [1752]
370 DATA 114BA5211DA50100,1E5           [800]
380 DATA 81CDE0BC2115A501,3C6           [886]
390 DATA 00A5CDD1BCF18721,498           [1110]
400 DATA 7230856F7E328B30,301           [1322]
410 DATA 237E328C30217830,258           [617]
420 DATA 7EB7283423CD5ABB,396           [1119]
430 DATA 18F6203831363234,233           [566]
440 DATA 0C4255464645522D,1F3           [1208]
450 DATA 494D5052494D414E,25D           [837]
460 DATA 54452030304B0D0A,17B           [1496]
470 DATA 454E20464F4E4354,22D           [940]
480 DATA 494F4E0D0A0A0A00,111           [1128]
490 DATA 21EDB02240003EC9,327           [1190]
500 DATA 3242002100301101,D7            [1146]
510 DATA 3001E80377FBC340,391           [704]
520 DATA 0008A5C36FA5C3A5,3EC           [1446]
530 DATA A54255464645D242,321           [1240]
540 DATA 434C4541D2000000,1E7           [1118]
550 DATA 0000B830B8300000,1D0           [1010]
560 DATA 0000000000000000,0             [964]
570 DATA 0000000000002DA5,D2            [1207]
580 DATA F32A2DA5221BA522,2F3           [1571]
590 DATA 19A51100A5EBA7ED,3F3           [1284]
600 DATA 524D440B6B621336,204           [1201]
610 DATA FFEDB0C9F30605C5,528           [1335]
620 DATA ED5B19A51AFEFF28,445           [899]
630 DATA 14CD32A6300F3EFF,335           [713]
640 DATA 12CDF3A5ED5319A5,475           [736]
650 DATA C110E4FBC9C1FBC9,5FE           [1548]
660 DATA FE01C0DD7E00B728,3F9           [701]
670 DATA 1AF32162A61128BD,32C           [1073]
680 DATA 010C00EDB0114BA5,2AB           [1316]
690 DATA 211DA5010081CDE0,312           [1271]
700 DATA BCFBC92156A61128,3D6           [1326]
710 DATA BD010C00EDB0211D,2A5           [659]
720 DATA A5CDE6BCFBC9B7C0,64F           [899]
730 DATA C32FA5E5D5C52162,499           [1167]
740 DATA A61128BD010C00ED,296           [199]
750 DATA B0C1D1E1C9F3E56F,633           [980]
760 DATA E5D5C5ED5B1BA51A,4A1           [1058]
770 DATA FEFFC2EBA57D2100,4ED           [920]
780 DATA A6460405280623BE,204           [754]
790 DATA 2320F87E12CDF3A5,430           [1029]
800 DATA ED531BA537C1D1E1,4AA           [1151]
810 DATA 7DE1FBC9A7C1D1E1,63C           [1004]
820 DATA 7DE1FBC9133E0093,406           [1285]
830 DATA 3EA59AC0ED5B2DA5,457           [1493]
840 DATA C90AA05EA15CA27B,3EB           [1992]
850 DATA A323A640AB7CAC7D,3FC           [1187]
860 DATA AD7EAE5DAF5BF3E5,518           [1651]
870 DATA 6FE5D5C5ED5B1BA5,4F6           [1123]
880 DATA 1AFEFF20BF18C5F3,4C6           [1635]
890 DATA E56FE5D5C5ED5B1B,536           [1674]
900 DATA A518A9CD4BA63002,356           [1138]
910 DATA 3FC9C506EFE67FED,514           [1541]
920 DATA 79F680ED79E67FED,5A7           [1387]
930 DATA 79C137C9C54F06F5,449           [615]
940 DATA ED78171779C1C900,396           [638]
950 DATA 0000000000000000,0             [964]
960 DATA 000000C3AAA5C3BC,391           [1348]
970 DATA A5C315A6C326A600,3B2           [950]
980 DATA *                              [109]
```

JESSICO

Quand les prix sont si bas, les souris dansent !

BOITIERS DE RANGEMENT

Boîtier Proto	: 10x3"	59 F
Boîtier DS40LA	: 30x3"	99 F
Boîtier DS40LB	: 45x3.50"	99 F
Boîtier JSY 80	: 80x3.50"	119 F
Boîtier JSY 48	: 48x3"	119 F
Boîtier DD50L	: 50x5.25"	119 F
Boîtier DS100L	: 100x5.25"	149 F



JSY 48 **119F**
avec serrure + clé
sauf Proto

JOYSTICKS

Les meilleurs joysticks au monde

Doubleur de Joystick	69 F
Phasor one (U.S. Gold)	115 F
Joystick Cheetah 125+	89 F
Joystick Cheetah Mach 1+	135 F
Pro 5000	130 F
Quickshot turbo II	129 F
Speed king	115 F
Switchjoy SJ.1	85 F
Switchjoy SJ.22	95 F
Wico 3 WAY	310 F
Wico the Boss	160 F
Cheetah PC	199 F
Carte PC	225 F

Gratuit : un doubleur de Joystick, pour l'achat de 2 phasor one ou 2 Cheetah mach I

RUBANS POUR IMPRIMANTES

Ruban CITIZEN 120D	79 F
Ruban DMP 2000	69 F
Ruban DMP 3000	69 F
Ruban DMP1	69 F
Ruban PCW 8256/8512	69 F

PERIPHERIQUES

Crayon Optique Dart	345 F
Digitaliseur Dart DMP2000	750 F
Interf. RS 232 MERCI	890 F
Interf. Série/PAR.PCW	890 F
Interf. RS232 CPC	590 F
Adapt. Péritel MP-2F-CPC	450 F
Souris AMX CPC	690 F
Souris Cameron CPC (D)	649 F
Synthét. Musical TMPI	980 F
Synthét. Vocal TMPI	545 F

dk'tronics

Ext. Mémoire 256K/464	785 F
Ext. Mémoire 256K/6128	785 F
Ext. Mémoire 512K/PCW	545 F
Ext. Mémoire 64K/464	475 F
Silicon Disc 256K/464	785 F
Silicon Disc 256K/6128	785 F

ACCESSOIRES

Protégez vos yeux

Copy Holder	189 F
Etiquet. 89x36 Par 500	69 F
Filtre Ecran 12" Coul.	179 F
Filtre Ecran 12" Monoc.	169 F
Support Imprimante 80 Col.	149 F
Support Moniteur 12-14"	139 F
Mouse Mat (tapis)	75 F



HOUSSES POUR AMSTRAD

Protégez votre ordinateur

Housse CPC 464 Coul.	109 F
Housse CPC 464 Monoch.	99 F
Housse CPC 6128 Coul.	109 F
Housse CPC 6128 Monoch.	99 F
Housse DISC FD1	65 F
Housse IMP. DMP 2000	79 F
Housse IMP. DMP 3000	79 F
Housse PC 1512 Coul.	119 F
Housse PC 1512 Mono.	105 F
Housse PCW 8256/8512	129 F
Housse Souris	59 F

CABLES D'EXTENSION

Ne restez plus collé à votre ordinateur !

Centr. Amstr./Imprim. Azerty	149 F
Câble Centr. Ams/Impr.	149 F
Câble Ecran/Clavier	149 F
Câble Extens. Joystick	65 F
Câble Extens. Port	170 F
Câble FD1	139 F
Câble Magnéto Cassette	55 F

DISQUETTES 3" pour AMSTRAD



15 F 90* à l'unité - Par 100

Réf.	10	20	50	100
3" CF2	*189 F*	*358 F*	*845 F*	*1590 F*
3" 1/2 DFDD	125 F	240 F	550 F	995 F
5" 1/4 SFDD	75 F	140 F	325 F	600 F
5" 1/4 DFDD	80 F	150 F	350 F	680 F

** Disquettes certifiées 100% garantie 5 ans*
Livrées dans boîtes + enveloppes + étiquettes

BOITIER PLASTIQUE type AMSOFT...... 30 F Les dix

UTILITAIRES

CPC	C D
AGENDA	150/165
ANALYSE FINANCIERE	*240*
AUTOFORM. ASSEMBLEUR	295
BUDGET (HIT 4)	155/190
C.A.O.	320/399
CALCUMAT	390
CARNET D'ADRESSES	150/165
D BASE 2	*760*
DATAMAT	425
DISCOLOGY	350
DR DRAW	625
DR GRAPH	625
GESTION DE FICHES	180/195
GESTION DOCUMENTAIRE	160/175
GESTION DOMESTIQUE	180/195
GESTION FAMILIALE	160/290
MASTERCALC	350
MASTERFILE	370
MULTIFACE 2	520
MULTIPLAN	478
POCKET WORDSTAR	825
SEMABANK	380
SEMASTATS	380
SEMFICH	360
TAS-SIGN	320
TASPRINT	275
TASWORD	380
TEXTOMAT	380
TURBO TUTOR	325
PCW	
ALIENOR	*960*
ANALYSE FINANCIERE	240
AUTOFROM. ASSEMBLEUR	295
D BASE 2	760
DATAMAT	395
DR. DRAW	625
DR. GRAPH	625
GESTION DOMESTIQUE	240
GESTION FICHIERS	250
MASTERFILE 8000	530
MULTIPLAN	478
POCKET BASE	720
POCKET WORDSTAR	825
QUICK MAILING	490
TAS-SIGN 8000	350
TASPRINT 8000	275
TASWORD+"MAIL MERGE"	410
WORDSTAR	890
PC	
FLIGHT SIMULATOR	399
MULTIPLAN JUNIOR	559
SIDEKICK	350
TAS-SIGN PC	370
TASPRINT PC	*370*
TASWORD+"MAIL MERGE"	420
EDUCATIFS CPC	
COURS DE BASIC	225
DECOUVERTE DE LA VIE	179
ECRIRE SS FAUTES V.1	*189*
ECRIRE SS FAUTES V.2	189
ENIGME A MADRID	225
ENIGME A MUNICH	225
ENIGME A OXFORD	225
EQUATIONS	195
FRANCAIS-CM	*195*
GEOMETRIE	195
MATHS 2E CYCLE 1	245
MATHS 2E CYCLE 2	195
MATHS 3E	195
MATHS 5E+4E	195
MATHS 6E	*195*
MATHS-CM	195
OBJECTIF EUROPE 4-3E	179
OBJECTIF FRANCE 4-3E	179
OBJECTIF MONDE 6E	179
VISA PR HYDE PARK 6E	195
LIBRAIRIE	
102 PROGRAMMES CPC	120
AMSTRAD EN FAMILLE	120
BIEN DEBUTER AVEC CPC	99
CLEFS AMSTRAD PCW	215
GRAND LIVRE DU BASIC 6128	149
GRAND LIVRE DU PCW	179

AMSTRAD TOP HITS

720 DEGRES	92/142	LA GUERRE DES ETOILES	96/146
ACE 2	92/142	LA MARQUE JAUNE	246/246
ADV.T.FIGHTER	87/136	LA MASCOTTE	192
ALBUM DIGITAL	116/162	LAZER TAG	92/142
ALBUM EPYX	116/186	LE MAITRE DES AMES	186
ALBUM HEWSON	92/142	LES DIEUX DE LA MER	142/192
ALBUM LORICIEL	92/142	LES MAITRES DE L'UNIVERS	92/142
ALBUM UBISOFT	162/192	LES RIPOUX	122/192
ALTERN WORLD GAMES	92/142	LORICIEL HIT 6	126/176
AMSTRAD GOLD HITS 2	92/142	MACH 3	126/176
ANNALES DE ROME	176/232	MAD BALLS	86/142
BAD CAT	92/142	MALETTE JEUX FIL	192/242
BARBARIAN	*92/132*	MASK 2	92/142
BASIL THE DETECTIVE	92/142	MATCH DAY 2	87/142
BASKET MASTER	87/142	MEURTRES EN SERIES	256/296
BEST OF ELITE 2	92/142	MEWILO	192
BILLY 2	132/192	NAVY MOVES	87/142
BIVOUAC	122/192	OCEAN STAR HIT 1	92/142
BLOOD VALLEY	*92/142*	*OCEAN STAR HIT 2*	*116/162*
BLUEBERRY	192	OUT RUN	92/142
BOB WINNER	126/166	OXPHAR	202
BOBSLEIGH	92/142	PACK FIL 2	136/186
BRAVESTAR	92/142	PEPE BEQUILLE	142/192
BUBBLE BOBBLE	92/142	PEUR SUR AMYTIVILLE	126/166
BUGGY BOY	*92/142*	*PHANTOM CLUB*	*87/142*
CALIFORNIA GAMES	92/142	PHANTYS	87/142
CAPTAIN AMERICA	92/142	PIRATES	92/142
CARRE D'AS	192	PLATOON	87/142
CATCH 23	87/142	PREDATOR	87/142
CHAMPIONSHIP SPRINT	92/142	PROFESSION DETECTIVE	226
CHOSE DE GROTEMB.	*132/172*	*PROHIBITION*	*117/192*
CLASH	192	PSYCHOSOLDIER	86/142
CLEVER AND SMART	132/176	QUAD	142/176
COMBAT SCHOOL	87/142	QUIN	242
CONFLICT 1	132/176	RAMPART	92/142
CRASHGARET	192	RASTAN	87/142
CRAZY CARS	126/156	RENEGADE	87/142
DEATH OR GLORY	96/142	ROAD RUNNER	92/142
ELITE 6 PACK N°2	92/145	ROLLING THUNDER	92/142
EXIT	142/192	RYGAR	92/142
F15 STRIKE EAGLE	96/146	SANTA FE	136/172
FER ET FLAMME	246	SEPTEMBER	92/142
FIRE TRAP	142/142	SHACKLED	92/142
FLYING SHARK	96/142	SIDE ARMS	92/142
GABRIEL	136/172	SILENT SERVICE	85/136
GAME SET MATCH	126/176	SKAAL	142/192
GAUNTLET 2	92/142	SORCERER LORD	176/232
GRYZOR	87/142	STREETSPORT BASKET	92/142
GUILD OF THIEVES	192	STRYFE 2	132/172
GUNSHIP	192/242	SUPER HANG ON	92/142
GUNSMOKE	92/142	SUPER SKI	142/192
HANS D'ISLANDE	126/196	THE LAST NINJA	92/142
HIT PACK 1	82/142	THE SENTINEL	96/146
HIT PACK 2	92/142	THEY SOLD A MILLION 3	92/142
HURLEMENTS	132/172	THUNDERCATS	87/142
IMAGINE ARCADE HITS	116/162	TOUR DE FORCE	92/142
IMPOSSIBLE MISSION 2	92/142	TRANTOR	92/142
INDIANA JONES	92/142	TRESORS DE US GOLD	116/192
INFILTRATOR 2	92/142	V 2	172
INTERNATIONAL KARATE 2	92/142	VICTORY ROAD	86/142
IZNOGOUD	192/242	WARLOCK	92
KONAMIS GREATEST HITS	92/142	WARLOCK	142
L'ANGE DE CRISTAL	142/202	WESTERN GAMES	132/176
L'ANNEAU DE ZENGARA	142/186	WIZARD WARZ	92/142
L'OEIL DE SET	142/186	WORLD LEAD.TOURN.	92/142

SUPER BUDGET K7 49 F

ARKANOID	EMPIRE	MARIO BROTHERS	SPY VS SPY 3
AVENGER	FIRELORD	MGT	TAI PAN
BALLBLAZER	FREDDY HARDEST	MINDSHADOW	TEMPEST
BLADE RUNNER	*FUTURE KNIGHT*	*MONTY ON THE RUN*	*THAI BOXING*
BREAKTHRU	GALVAN	MOVIE	THANTOS
BRUCE LEE	GAME OVER	NEMESIS	THUNDERBIRDS
CAULDREN 2	HIJACK	NEMESIS THE WARLOCK	TOP GUN
CHIMERA	HIVE	NEXUS	TRAILBLAZER
CHOLO	HOBBIT	PRODIGY	WARLOCK
DANDY	*JAILBREAK*	*RANARAMA*	*WIZBALL*
DONKEY KONG	JET SET WILLY	ROCCO	XARO
DRAGONSLAIR	KRAKOUT	SIGMA 7	YIE AR KUNG FU
EIDOLON	KWAH	SPY VS SPY	YIE AR KUNG FU II

BON DE COMMANDE *EXPRESS*

NOUVEAU : Boutique à Nice "**HOLLYWOOD STARS**" 8, Bd Joseph Garnier
à retourner à **JESSICO IMPEX - BP 693 - 06012 NICE CEDEX**

ARTICLE (garantie échange immédiat)	Qte	PRIX	MONTANT
DOM-TOM + ETRANGER 50F	TOTAL des COMMANDES		
	FRAIS DE PORT*		25
	TOTAL A REGLER		

port logiciel jeux 15 F

☐ Je joins un chèque ou mandat-lettre
☐ *Je paye par carte bleu et je complète les 2 lignes ci-dessous*

carte bleue ....
date d'expiration ....
NOM .... PRENOM ....
N° ET RUE ....
VILLE .... CODE POSTAL ....
Signature obligatoire

GAGNEZ DU TEMPS ! **Commandez par téléphone : 93 51 61 30**

Précisez pour quel ordinateur :

Disc ☐ K7 ☐

Budget suivant stock disponible

REPRO-JM - NICE

REPERAGE D'ERREURS DANS UN PROGRAMME

Cet article s'adresse à ceux d'entre vous qui passent des heures à taper sur leur CPC un programme qu'ils trouvent dans une revue comme celle que vous avez entre les mains, ou qu'ils ont conçu eux-mêmes, pour obtenir un superbe plantage du système ou un mauvais fonctionnement du programme à la suite d'une faute de frappe ou d'une réelle erreur dans le programme d'origine.

Le problème de compatibilité entre les machines

La première chose à regarder lorsque vous tapez un programme provenant d'une revue est de vérifier que celui-ci tourne bien sur votre ordinateur. Il est en effet inutile de retaper tel quel sur un CPC un programme prévu pour une APPLE ! Il est obligatoire de le trancrire car le Basic, malgré le fait qu'il soit le langage utilisé par la plupart des micro-ordinateurs familiaux du marché, est différent d'une machine à l'autre.

Le problème se pose également entre les trois modèles CPC lorsqu'il s'agit d'un programme utilisant certaines routines se trouvant en ROM ou encore certaines variables systèmes. Vérifiez dans ce cas que le programme qui vous intéresse tourne bien sur votre modèle de CPC (si bien sûr, vous en possédez un !).

Les programmes utilisant le langage machine

Certains programmes de jeux ou utilitaires possèdent une partie en langue machine. Pour que vous puissiez les retaper sans avoir à utiliser d'assembleur, les codes opérations (ces codes sont les seules instructions comprises par le microprocesseur, cerveau d'un micro-ordinateur, qui correspondent à des fonctions élémentaires tels que chargement ou sauvegarde d'un registre) sont mis en DATA et le plus souvent codés en hexadécimal.

Lorsque vous avez fini de taper le programme, la première chose à faire est de le SAUVEGARDER. C'est impératif ! Si une des valeurs est erronée, lors d'un RUN, vous risqueriez de planter le programme et il vous faudrait alors retaper entièrement le programme car vous ne pourriez pas le récupérer.

Après cette sauvegarde, vous pouvez le lancer. S'il plante, ce n'est pas grave puisque vous l'avez sur cassette ou sur disquette. Vous n'avez donc pas à le retaper (avouez qu'il est rageant de taper un programme pendant deux heures pour rien !). Rechargez-le alors en mémoire et vérifiez-le lignes par lignes, instructions par instructions, et même caractère par caractère. Le langage machine est impitoyable dans le cas d'une erreur. Le seul message qu'il renvoie est le plantage pur et simple du micro-ordinateur. Soyez donc très prudent.

Si, pour vos programmes, vous employez le système des DATA pour rentrer du langage machine et surtout, si vous désirez nous envoyer un programme utilisant ce procédé, forcez-vous à inclure dans le programme une routine d'auto-test qui, justement, vérifiera que les valeurs en DATA sont correctes. Voici une solution parmi tant d'autres. Il suffit d'utiliser une variable supplémentaire qui contiendra au fur et à mesure de l'implantation en mémoire des valeurs lues en

DATA, la somme de celles-ci. En supposant que la routine de decture des DATA se trouve en début de programme, voici un exemple que nous vous conseillons car elle évite le plantage fatal en cas d'erreur de recopie du programme :
10 somdat = O' Initialisation de la variable test
20 FOR ad = *adresse de début* TO adresse de fin
30 READ da : POKE ad,da :somdat = somdat + da

40 NEXT ad
50 IF somdat < > X THEN ?"ERREUR DANS LES DATA" : END
60 suite du programme
X est la valeur théorique qui est égale à la somme de toutes les valeurs contenues dans les lignes de DATA. Si le message "ERREUR DANS LES DATAS" s'affiche, c'est que vous avez fait une faute de saisie, ou que la valeur de X est fausse. De toutes façons, vérifiez vos DATA.
Si c'est un message d'erreur du BASIC qui s'affiche, c'est qu'une donnée est incorrecte, ou qu'il manque une virgule... Pour connaître la DATA défaillante, faites simplemen : PRINT ad — adresse de début.
Le nombre qui sera alors affiché sera le numéro de la DATA qui est incorrecte. Il ne vous reste plus qu'à les compter jusqu'à ce que vous pointiez sur celle qu'il vous faut modifier. Cette dernière astuce est valable pour tous les programmes BASIC contenant des DATA lorque l'écriture de ces dernières provoque un message d'erreur lors d'un RUN.

Les messages d'erreurs du basic

Ces messages sont écrits en anglais du fait de la nationalité de l'AMSTRAD et de celle du BASIC. Malgré tout, le manuel du CPC vous dit succintement ce qu'ils signifient. Reportez-vous y de tempms en temps si vous ne comprenez pas votre message d'erreur, vous trouverez peut-être quelle en est la cause.
L'erreur la plus sournoise est le MEMORY FULL, qui signifie mémoire pleine. Elle peut survenir lors d'un SYMBOL, AFTER prenant trop de place en mémoire, ou encore lors d'un MEMORY. Elle peut aussi survenir lors d'une instruction comme DIM si la réservation de mémoires variables demandée est trop importante. Vous pouvez toujours connaître la taille mémoire disponible en tapant PRINT FRE(O). Le résultat obtenu sera en octets. Pour connaître la valeur en Ko, il vous suffit de diviser ce nombre par 1 024 ($= 2^{10}$). Ce message d'erreur peut avoir encore une origine beaucoup plus sournoise. Si vous avez trop de boucles imbriquées, ou de GOSUB ou de WHILE, vous obtiendrez un MEMORY FULL. Vérifiez donc que vous n'avez pas de sortie de boucles ou de sous-programmes, anormaux. Dans ce cas, comme dans toutes les recherches d'erreurs, nous vous conseillons d'utiliser l'instruction TRON qui vous affichera, au fur et à mesure, les numéros des lignes parcourues par le programme. Vous pourrez donc ainsi vérifier que tout se passe bien comme prévu au niveau des lignes.

Correction d'une erreur de syntaxe

Ce type d'erreur est signalé, à l'exécution du programme, par le message SYNTAX ERROS IN LINE XXXX. C'est le type d'erreur le plus simple à corriger. Vérifiez que les instructions sont bien écrites. A ce propos, un conseil, lorsque vous tapez un programme, faites-le en minuscules. L'interpréteur convertira automatiquement les instructions pour qu'elle apparaissent en majuscules dans votre programme. Ainsi, si dans une ligne, une instruction n'apparaît pas en majuscules, c'est que sa syntaxe est incorrecte. Si toutes les instructions sont correctes, vérifiez alors que vous avez bien laissé un espace entre une instruction BASIC et son, ou ses, paramètres. Par exemple dans le cas de BORDER O, il faut obligatoirement un espace entre BORDER et O. Vérifiez aussi que vous avez bien mis les : entre deux instructions. s'il s'agit d'un programme que vous êtes en en train de créer, n'utilisez pas de variables dont le nom contient celui d'une instruction appartenant au langage que vous utilisez. De plus, n'oubliez pas les paramètres nécessaires à certaines fonctions. N'hésitez pas, en cas de doute, à vous reporter au manuel de votre ordinateur, il est là pour ça !

Autres messages d'erreur

Certains messages d'erreurs, tel un SUBSCRIPT OUT OF RANGE IN LINE XXXX, ne vous donneront pas le numéro de la ligne où s'est réellement produite l'erreur. Dans le cas du SUBSCRIPT OUT OF RANGE, vous avez oublié de dimensionner votre variable (ou l'indice de dimension était trop petit) par l'instruction DIM.

Les erreurs anonymes

C'est le type d'erreurs le plus difficile à détecter, la bête noire ees programmeurs. Le programme ne fonctionne pas correctement, mais aucun message d'erreur ne s'affiche. Le cas des DATA, vu précédemment, en est un bon exemple. Ne confondez pas un I avec un i ou un 1, ou encore un O avec un 0. Un programme fait ce qu'on lui demande, à condition que cela soit fait dans les normes. La programmation est quelque chose de très strict. soyez donc très vigilant !
Vous avez certainement remarqué, en lisant notre revue, des nombres entre crochets à la droite de chaque ligne des listings. Ils ne font, bien sûr, pas partie des programmes, mais sont utilisés pour le debugging (terme du jargon informatique signifiant recherche d'erreur) en utilisant le programme vérificateur V2 (Amstrad Magazine n° 16 p. 88) que nous republierons régulièrement. Ce dernier doit être utilisé dès que vous tapez un programme de notre revue. Il vous signalera lorsque vous entrez une ligne de programme si elle contient ou non, une erreur. Pratique n'est-ce pas ?
De toutes façons, la meilleure façon de savoir corriger une erreur dans un programme, est encore de savoir programmer. Une très bonne méthode pour apprendre la programmation de base lorsque l'on est seul, est de transcrire un programme simple en BASIC d'un ordinateur à un autre. Cela impose une connaissance croissante du BASIC de son ordinateur ainsi qu'une méthode de recherche performante qui est utile lors du debugging. Ceci dit, bonne programmation !

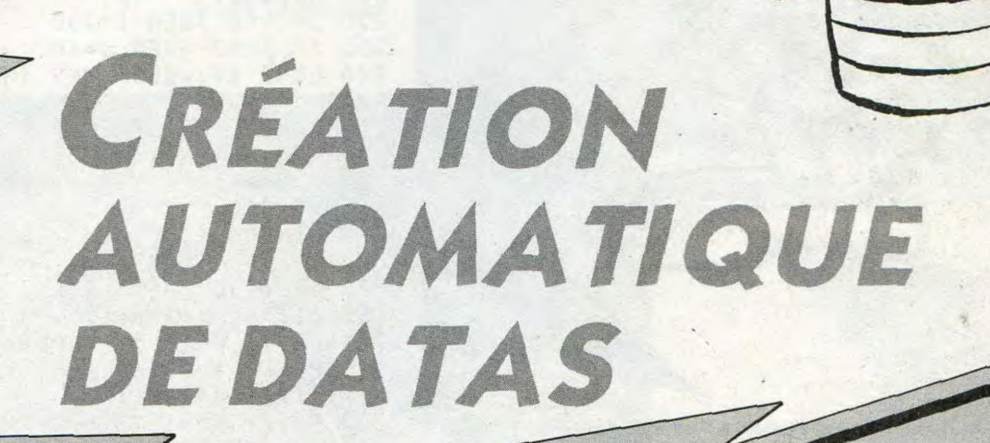
Eric Mistelet

A titre d'exercice

Le programme vérificateur V2 paru dans Amstrad Magazine n° 16 fonctionne très bien pour les trois modèles de CPC. S'il ne fonctionne pas correctement, c'est que vous avez commis une faute de frappe quelque part. Nous vous rappelons que vous devez sauvegarder le programme avant de le lancer par un RUN. En effet, une erreur dans les DATA pourrait causer un plantage du système.
Nous vous proposons donc une petite modification qui vous signalera une éventuelle erreur dans ces DATA. ATTENTION, tapez bien le programme prévu pour votre version de CPC, 464, 664, 6128. Dans la modification que nous vous proposons, la variable SN dépend, elle aussi, du modèle de CPC que vous possédez.
Lignes à rajouter au vérificateur V2 :
Pour les trois modèles :
135 somme = somme + VAL("&" + byte$)
155 IF some < > sn THEN PRINT "ERREUR DANS LES DATA" : END
— Version CPC 464 : 115 somme = O :sn = 33109
— Version CPC 664 : 115 somme = O :sn = 33249
— Version CPC 6128 : 115 somme = 0 :sn = 33203
Si, lorsque vous lancez le programme, le message ERREUR DANS LES DATA s'affiche, vérifiez vos DATA , sauvegardez votre programme après que les modifications nécessaires aient été effectuées. Nous vous rappelons que les commandes du vérificateur V2 sont : |ON; |OFF; |CHECK,2 (|CHECK,8 pour imprimante).

NOTA : la barre verticale («|») s'obtient en appuyant sur <shift> et @.

CRÉATION AUTOMATIQUE DE DATAS



Un fichier binaire destiné à la prospérité (par l'intermédiaire d'une revue par exemple) se doit d'être transformé en une suite de codes héxadécimaux, inclus dans un programme chargeur Basic. Ce dernier, signalant les erreurs de saisie éventuelles, a pour mission d'implanter les données en mémoire, puis de réaliser une sauvegarde automatique du langage machine sur cassette ou disquette. Ce travail de création accompli, il peut être supprimé.
Outre la métamorphose d'un fichier binaire en datas Basic, l'ouvrage ici proposé, utilisant le lecteur de disquettes, offre le programme chargeur spécifique, nanti d'un REM précisant le nom de la création binaire.

Chargez ce programme par LOAD, placez la disquette contenant le ou les fichiers binaires à transformer et faites RUN.

Le programme effectue un catalogue de tous les fichiers d'extension « .BIN » figurant sur la disquette, puis réclame le nom du fichier à transformer, son adresse de chargement (&A000 par défaut) et son adresse de fin. Une validation sans spécification, réclame alors la longueur du fichier. Le choix du canal de sortie (écran, imprimante ou disquette) est ensuite demandé (écran par défaut), ainsi que le numéro de la première ligne de datas (100 par défaut) et leur pas de numérotation (10 par défaut). Dans le cas d'une sortie sur disquette, le nom du programme chargeur Basic est évidemment à préciser.
Nota : Les adresses de chargement et de fin, ainsi que la longueur du fichier, peuvent être rentrées en décimal, ou en hexadécimal (avec le préfixe « & »).

Stéphane Schreiber

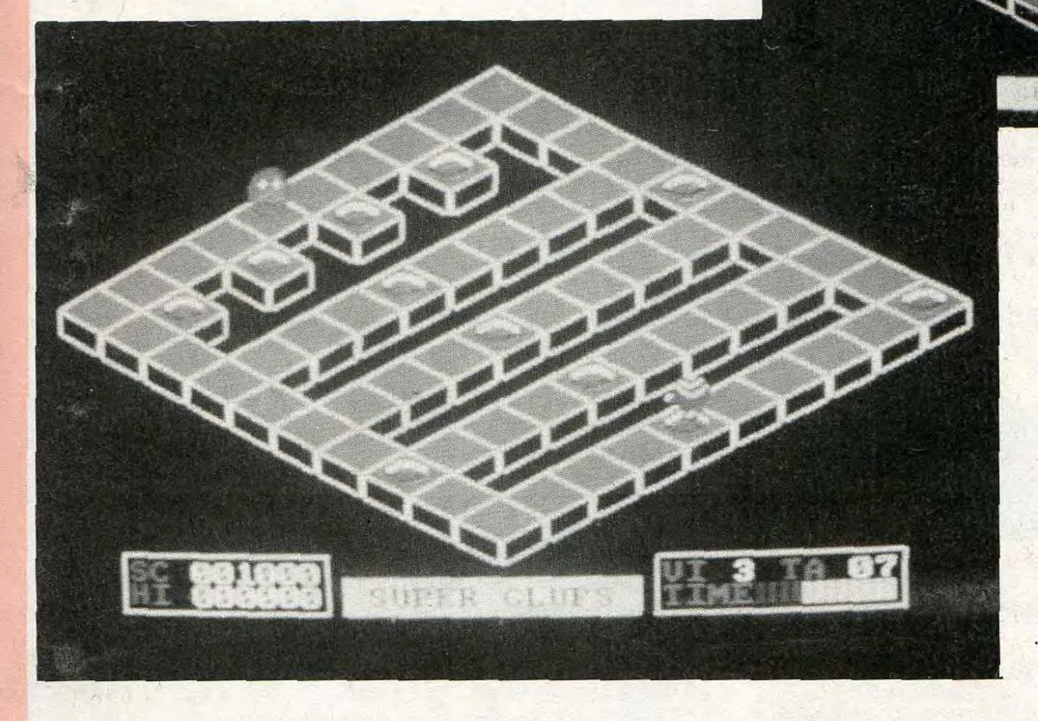
CRÉATION AUTOMATIQUE DE DATAS

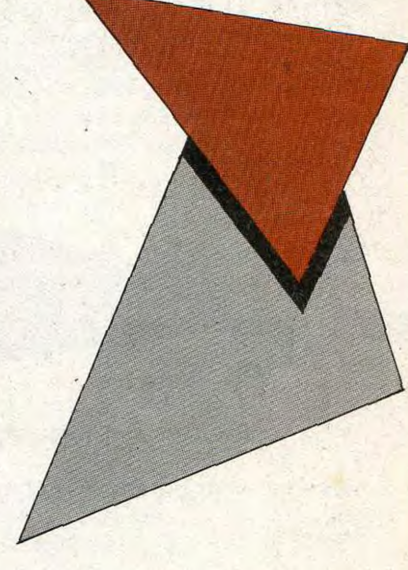
```
10 ' ******************************** [1252]
20 ' *     Creation automatique     * [1840]
30 ' *          de   datas         * [634]
40 ' *   par Stephane SCHREIBER     * [657]
50 ' ******************************** [1252]
60 IF HIMEM<>&3FFF THEN OPENOUT "se [3332]
pth":MEMORY &3FFF:CLOSEOUT
70 MODE 2:dir$="*.bin":|DIR,@dir$   [2111]
80 INPUT "Programme binaire ? ",a$  [2948]
90 IF a$="" THEN CLS:CAT:GOTO 80    [1122]
100 a$=UPPER$(a$):IF INSTR(a$,".")= [2308]
0 THEN a$=a$+".BIN"
110 INPUT "Adresse de chargement (& [2791]
A000 par defaut) ? ",ad
120 IF ad=0 THEN ad=&A000           [1687]
130 IF a<0 THEN a=65536+a           [1045]
140 INPUT "Adresse de fin ? ",af    [1851]
150 IF af=0 THEN PRINT STRING$(2,11 [3182]
):INPUT "      Longueur ? ",long ELS
E 170
160 IF long=0 THEN PRINT CHR$(11):G [3621]
OTO 150 ELSE af=ad+long
170 PRINT:INPUT "Canal de sortie: < [4662]
E>cran, <I>mprimante ou <D>isk (Ecr
an par defaut) ? ",q$:q$=UPPER$(q$)
180 IF q$="" OR q$="E" THEN q=0 ELS [4970]
E IF q$="I" THEN q=8:WIDTH 40 ELSE
IF q$="D" THEN q=9 ELSE PRINT STRIN
G$(3,11):GOTO 170
190 IF q=9 THEN INPUT "    Nom du fi [3268]
chier ? ",nom$:IF INSTR(nom$,".")=0
 THEN nom$=nom$+".BAS"
200 PRINT:INPUT "Premiere ligne de  [4261]
numerotation (100 par defaut) ? ",l
210 INPUT "Pas de numerotation (10  [2868]
par defaut) ? ",p
220 IF l=0 THEN l=100               [587]
230 IF p<=0 THEN p=10               [1092]
240 LOAD a$,ad:IF q=9 THEN OPENOUT  [1636]
nom$
250 MODE 2:PRINT CHR$(24);"Prg.: "; [5361]
a$;", de &";HEX$(ad);" a &";HEX$(af
);".  Premiere ligne:";l;".  Pas:";
p;CHR$(24)
260 WINDOW 1,80,3,25:PAPER 0:PEN 1: [2691]
CLS
270 IF q<>0 THEN LOCATE #1,33,12:PR [3523]
INT #1,"En travail..."
280 PRINT #q," 5 ' Createur de "+a$ [1602]
290 PRINT #q," 10 MEMORY &"+HEX$(ad [3081]
-1)+":l="+STR$(l)
300 PRINT #q," 20 FOR i=&"+HEX$(ad) [3742]
+" TO &"+HEX$(af)+" STEP 8"
310 PRINT #q," 30 s=0:FOR j=i TO i+ [1685]
7:READ a$"
320 PRINT #q," 40 c=VAL("+CHR$(34)+ [4001]
"&"+CHR$(34)+"+a$):POKE j,c:s=s+c:N
EXT"
330 PRINT #q," 50 READ s$:IF s<>VAL [8710]
("+CHR$(34)+"&"+CHR$(34)+"+s$) THEN
 PRINT "+CHR$(34)+"Erreur DATAs lig
ne"+CHR$(34)+";l:STOP"
340 PRINT #Q," 60 l=l+"+STR$(p)+":N [2242]
EXT"
350 PRINT #q," 70 SAVE"+CHR$(34)+a$ [5054]
+CHR$(34)+",b,&"+HEX$(ad)+",&"+HEX$
(af-ad)+",&"+HEX$(ad)
360 FOR i=ad TO af STEP 8           [575]
370 s=0:PRINT #q,l;"DATA ";:FOR j=0 [3093]
 TO 7
380 h$=HEX$(PEEK(i+j),2)            [1242]
390 PRINT #q,h$;",";                [1075]
400 s=s+VAL("&"+h$)                 [637]
410 NEXT j                          [370]
420 PRINT #q,HEX$(s,3):l=l+p        [1462]
430 NEXT i                          [375]
440 l=l+p:PRINT #q,l;"END"          [1579]
450 END                             [110]
```

SUPER GLUPS 3D

Comparable à l'univers Pac Man, le monde Super Glups s'enorgueillit d'une troisième dimension, conférant à l'intrigue habituelle un surcroît de difficulté. Les vingt tableaux modifiables de ce jeu, sont autant d'épreuves déroutantes pour le joueur humanoïde dénué du sens inné de l'orientation et déboussolé par l'implacable cruauté de Big Blue.
Sauvegardez le premier listing sous un nom de votre choix et le second (règles et programme principal), sous le nom « GLUP1 ».

Claude Lemoullec





```
10 EVERY 25,2 GOSUB 320                    [813]
20 MODE 1:INK 0,0:BORDER 0:INK 1,0:        [3903]
INK 2,2:INK 3,19
30 a$="L M C":x=200:y=340:GOSUB 240        [1666]
40 a$="SOFTWARE":x=125:y=250:GOSUB         [1634]
240
50 LOCATE 1,1:PRINT"              ":INK    [1229]
 1,6
60 LOCATE 17,16:PEN 3:PRINT"PRESENT        [1212]
S"
70 SYMBOL AFTER 32:DIM a(15)               [1699]
80 LOCATE 7,20:PEN 1:a$="SUPER GLUP        [2334]
S 3D"
90 FOR H=1 TO LEN(a$):FOR j=0 TO 7         [1993]
100 x=8*(ASC(MID$(a$,h))-32)               [1811]
110 x$=BIN$(PEEK(HIMEM+1+x+j),8)           [2566]
120 im$="":FOR k=1 TO 8                    [1677]
130 f$=MID$(x$,k,1)                        [1495]
140 im$=im$+f$+f$                          [797]
150 NEXT k                                 [373]
160 a(j*2)=VAL("&X"+LEFT$(im$,8))          [1872]
170 a(j*2+1)=VAL("&X"+RIGHT$(im$,8)        [1359]
)
180 NEXT j                                 [370]
```

```
190 SYMBOL 250,a(0),a(2),a(4),a(6), [2932]
a(8),a(10),a(12),a(14)
200 SYMBOL 251,a(1),a(3),a(5),a(7), [2119]
a(9),a(11),a(13),a(15)
210 PRINT CHR$(250);:PRINT CHR$(251 [2272]
);
220 NEXT h                          [372]
230 RUN "!GLUP1                     [680]
240 a=LEN(a$):a8=a*8                [1636]
250 LOCATE 1,1:PEN 1:PRINT a$;:PEN  [1944]
1
260 tx=x:y2=398                     [634]
270 FOR f=1 TO 8:x2=0:FOR g=1 TO a8 [1083]
280 IF TEST( x2,y2)=1 THEN PLOT x+4 [9399]
,y,2:PLOT x+4,y-2:PLOT x,y-2:PLOT x
+2,y-2:PLOT x+4,y-4:PLOT x+2,y-4:PL
OT x,y-4:PLOT x-2,y+2,3:PLOT x,y+2:
PLOT x+2,y+2:PLOT x+2,y:PLOT x-2,y:
PLOT x,y
290  x=x+6:x2=x2+2                  [1284]
300 NEXT g:y=y-6:tx=tx:y2=y2-2:x=tx [2056]
:NEXT f
310 RETURN                          [555]
320 DI:IF (SQ(1) AND 7)=0 THEN EI:R [5410]
ETURN ELSE READ p,d:IF p=-1 THEN RE
STORE 340:GOTO 320
330  SOUND 1,P,D*0.5:SOUND 2,0.5*P, [4958]
D*0.5:SOUND 4,0.25*P,D*0.5:GOTO 320
340 DATA 478,050,379,050,358,050,31 [2116]
9,200,000,005,319,050
350 DATA 478,050,379,050,358,050,31 [2122]
9,200,000,005,319,100
360 DATA 478,050,379,050,358,050,31 [2903]
9,100,379,100,478,100
370 DATA 379,100,426,200,000,005,42 [2799]
6,050,379,050,000,005
380 DATA 379,050,426,050,478,150,00 [2304]
0,005,478,050,379,100
390 DATA 319,100,000,005,319,050,35 [2988]
8,150,000,005,358,100
400 DATA 379,050,358,050,319,100,37 [2269]
9,100,478,100,426,100
410 DATA 478,200,000,005,478,050,00 [2465]
0,005,000,005,-01,001
```

```
10 BORDER 0:MODE 1:INK 0,0:INK 1,26 [1724]
:INK 2,6:INK 3,18:CLS
20 PEN 3:LOCATE 11,8:PRINT STRING$( [6438]
21,"*"):LOCATE 11,18:PRINT STRING$(
21,"*"):FOR h=9 TO 17:LOCATE 11,h:P
RINT "*":LOCATE 31,h:PRINT"*":NEXT
30 PEN 2:LOCATE 13,10:PRINT"1 -";:P [3311]
EN 1:PRINT" REGLES DE JEU"
40 PEN 2:LOCATE 13,13:PRINT"2 -";:P [3796]
EN 1:PRINT" JOYSTICK"
50 PEN 2:LOCATE 13,16:PRINT"3 -";:P [3116]
EN 1:PRINT" CLAVIER"
60 A$=INKEY$:IF a$="" THEN GOTO 60  [1836]
70 PRINT CHR$(7):IF A$="1" THEN GOT [2117]
O 110
80 IF A$="2" THEN GOTO 290          [949]
90 IF A$="3" THEN GOTO 300          [1449]
100 GOTO 60                         [390]
110 CLS:MODE 1:LOCATE 15,1:PRINT"SU [4937]
PER  GLUPS":PEN 2:LOCATE 15,2:PRINT
"------------"
120 PEN 3:LOCATE 1,4:PRINT"    Les  [7645]
jeux les plus simples sont les  mei
lleurs et les plus efficaces."
130 LOCATE 1,7:PRINT"      Un arene [6978]
de jeu,des couronnes qu'un";:PEN 1:
PRINT"SUPER GLUPS"
140 LOCATE 13,8:PEN 3:PRINT" doit r [2766]
amasser."
150 LOCATE 1,10:PRINT"      Tout en [4605]
evitant les contacts avec"
160 LOCATE 1,11:PEN 2:PRINT"BIG BLU [5507]
E";:PEN 3:PRINT" son ennemi jure.."
170 LOCATE 1,13:PRINT"     Un theme [7026]
 des plus classique pour unjeu qui
ne l'est pas"
180 LOCATE 1,16:PRINT"      Ici les [9265]
decors sont en 3D et les 20tableaux
 qui composent le jeu sont tous des
 pieges differents"
190 LOCATE 1,20:PRINT"      En REM v [7245]
ous trouverez le moyen pourcreer vo
s propres tableaux si les miens ne
suffisent pas."
200 LOCATE 33,25:PEN 1:PRINT"<ENTER [1706]
>":CALL &BB18
210 CLS:LOCATE 15,1:PRINT"DEPLACEME [5263]
NTS":PEN 2:LOCATE 15,2:PRINT"******
******"
220 LOCATE 1,4:PRINT"Joystick/clavi [5522]
er":PEN 1:LOCATE 1,5:PRINT"--------
--------"
230 LOCATE 8,7:PRINT CHR$(240):LOCA [4284]
TE 14,7:PRINT"=  avance du GLUPS"
240 LOCATE 8,10:PRINT CHR$(243):LOC [4932]
ATE 14,10:PRINT"=  1/4 de tour a dr
oite"
250 LOCATE 8,13:PRINT CHR$(242):LOC [7076]
ATE 14,13:PRINT"=  1/4 de tour a ga
uche"
260 PEN 3:LOCATE 6,19:PRINT STRING$ [2816]
(30,"=")
270 PEN 2:LOCATE 15,22:PRINT "BONNE [3025]
 CHANCE"
280 LOCATE 33,25:PEN 1:PRINT"<ENTER [1995]
>":CALL &BB18:GOTO 10
290 ga=74:da=75:ht=72:GOTO 360      [1246]
300 ga=8:da=1:ht=0:GOTO 360         [1292]
310 REM :::::::::::::::::::::::::   [1823]
320 REM :                       :   [419]
330 REM :    INITIALISATION     :   [1034]
340 REM :                       :   [419]
350 REM :::::::::::::::::::::::::   [1823]
360 SYMBOL AFTER 199                [1457]
370 SYMBOL 200,0,0,0,0,1,6,24,96    [1557]
380 SYMBOL 201,1,6,24,96,128,0,0,0  [1590]
390 SYMBOL 202,128,96,24,6,1,0,0,0  [2108]
400 SYMBOL 203,0,0,0,0,128,96,24,6  [2254]
410 SYMBOL 204,0,0,0,0,0,1,7,31     [1412]
420 SYMBOL 205,0,1,7,31,127,255,255 [2149]
,255
430 SYMBOL 206,0,128,224,248,254,25 [2500]
5,255,255
440 SYMBOL 207,0,0,0,0,0,128,224,24 [1466]
8
```

```
450 SYMBOL 208,127,31,7,1,0,0,0,0    [1612]
460 SYMBOL 209,255,255,255,255,127,  [2493]
31,7,1
470 SYMBOL 210,255,255,255,255,254,  [1638]
248,224,128
480 SYMBOL 211,254,248,224,128,0,0,  [2326]
0,0
490 SYMBOL 212,128,128,128,128,128,  [2400]
128,128,128
500 SYMBOL 213,1,1,1,1,1,1,1,1       [1355]
510 SYMBOL 214,0,0,0,0,7,31,120,96   [1838]
520 SYMBOL 215,0,0,0,0,224,248,30,6  [1810]
530 SYMBOL 216,96,120,31,7,0,0,0,0   [1950]
540 SYMBOL 217,6,30,248,224,0,0,0,0  [1948]
550 SYMBOL 218,0,12,12,28,31,63,60,  [1909]
60
560 SYMBOL 219,0,0,0,0,255,255,0,0   [1915]
570 SYMBOL 222,238,238,238,238,238,  [1976]
238,238,0
580 REM :::::::::::::::::::::::::    [1823]
590 REM :                       :    [419]
600 REM :      ROUTINES LM      :    [1948]
610 REM :                       :    [419]
620 REM :::::::::::::::::::::::::    [1823]
630 MEMORY &7FFF                     [150]
640 REM ::::: GLUPS  GAUCHE :::::    [1501]
650 RESTORE 660:FOR h=&9000 TO &900 [3602]
0+63:READ A$:POKE h,VAL("&"+a$):NEX
T
660 DATA 00,33,FF,88,00,77,FF,CC,00 [8895]
,76,FF,EE,00,EF,FD,EE,00,FE,DF,FF,1
1,FB,FD,FF,76,F7,FF,FF,F9,FF,FF,FF,
F9,FF,FF,FF,66,77,F6,EE,00,77,F9,CC
,00,33,FF,CC,00,00,CC,CC,00,11,99,8
8,00,11,99,88,00,77,33,00
670 REM ::::: GLUPS   BAS   :::::    [1946]
680 RESTORE 690:FOR h=&9041 TO &904 [4017]
1+63:READ A$:POKE h,VAL("&"+a$):NEX
T
690 DATA 11,FF,CC,00,33,FF,EE,00,77 [11180]
,FF,E6,00,77,FB,7F,00,FF,BF,F7,00,F
F,FB,FD,88,FF,FF,FE,E6,FF,FF,FF,F9,
FF,FF,FF,F9,77,F6,EE,66,33,F9,EE,00
,33,FF,CC,00,33,33,00,00,11,99,88,0
0,11,99,88,00,00,CC,EE,00
700 REM ::::: GLUPS   HAUT  :::::    [1581]
710 RESTORE 720:FOR h=&90C3 TO &90C [3539]
3+63:READ A$:POKE h,VAL("&"+a$):NEX
T
720 DATA 00,33,F3,88,00,73,FF,C4,00 [11313]
,FC,FC,EE,11,F7,F3,EC,11,FB,FF,FB,7
7,FC,FE,F7,F9,FF,F1,FF,FB,FF,FF,FF,
77,FF,FF,FF,00,FF,FF,EE,00,77,FF,CC
,00,33,FF,CC,00,11,99,88,00,33,33,0
0,00,77,33,00,00,EE,66,00
730 REM ::::: GLUPS DROITE  :::::    [1940]
740 RESTORE 750:FOR h=&9104 TO &910 [3432]
4+63:READ A$:POKE h,VAL("&"+a$):NEX
T
750 DATA 11,FC,CC,00,32,FF,EC,00,77 [10017]
,F3,F3,00,73,FC,FE,88,FD,FF,FD,88,F
E,F7,F3,EE,FF,F8,FF,F9,FF,FF,FF,FD,
FF,FF,FF,EE,77,FF,FF,00,33,FF,EE,00
,33,FF,CC,00,11,99,88,00,00,CC,CC,0
0,00,CC,EE,00,00,66,77,00
760 REM :::::   BIG BLUE   :::::     [1330]
770 RESTORE 780:FOR h=&9082 TO &908 [4841]
2+63:READ A$:POKE h,VAL("&"+a$):NEX
T
780 DATA 00,77,EE,00,11,8F,1F,88,23 [9697]
,0F,0F,4C,47,0F,0F,4C,47,4B,2D,2E,C
F,E1,78,2E,8F,4B,2D,1F,8F,0F,0F,1F,
8F,0F,0F,1F,8F,0F,0F,1F,8F,0F,0F,1F
,47,3F,CF,2E,47,4F,2F,2E,23,0F,0F,4
C,11,8F,1F,88,00,77,EE,00
790 REM ::::: AFFICHAGE SPRITE :::: [2488]
:
800 RESTORE 810:FOR h=&A000 TO &A05 [3764]
C:READ A$:POKE h,VAL("&"+a$):NEXT
810 DATA DD,5E,00,DD,56,01,DD,6E,02 [7395]
,DD,66,03,06,10,CD,19,BD,F3,C5,E5,0
6,04,1A,E6,88,28,06,4F,7E,E6,77,B1,
77,1A,E6,44,28,06,4F,7E,E6,BB,B1,77
,1A,E6,22,28,06,4F,7E,E6
820 DATA DD,B1,77,1A,E6,11,28,06,4F [6561]
,7E,E6,EE,B1,77,23,13,10,D0,E1,CD,5
1,A0,C1,10,C5,FB,C9,00,00,7C,C6,08,
67,D0,D5,11,50,C0,19, D1,C9
830 REM ::::: ENLEVE SOUS SPRITE :: [1722]
::
840 RESTORE 850:FOR h=&A060 TO &A07 [3093]
C:READ A$:POKE h,VAL("&"+a$):NEXT
850 DATA 11,00,A4,DD,6E,00,DD,66,01 [5868]
,06,10,C5,E5,06,04,7E,12,13,23,10,F
A,E1,CD,51,A0,C1,10,EF,C9
860 REM ::::: REMET SUR SPRITE :::: [2038]
:
870 RESTORE 880:FOR h=&A080 TO &A09 [3495]
C:READ A$:POKE h,VAL("&"+a$):NEXT
880 DATA 11,00,A4,DD,6E,00,DD,66,01 [5080]
,06,10,C5,E5, 06,04,1A,77,23,13,10,
FA,E1,CD,51,A0,C1, 10,EF,C9
890 FOR h=&A060 TO &A07C:a=PEEK(h): [4784]
POKE h+&100,a:NEXT:POKE &A161,&50
900 FOR h=&A080 TO &A09C:a=PEEK(h): [2943]
POKE h+&100,a:NEXT:POKE &A181,&50
910 REM :::::::::::::::::::::::::    [1823]
920 REM :                       :    [419]
930 REM :   VARIABLES DE BASE   :    [2081]
940 REM :                       :    [419]
950 REM :::::::::::::::::::::::::    [1823]
960 CALL &BBFF:INK 0,0:INK 1,26:INK [3748]
 2,2:INK 3,6:BORDER 0:CLS
970 c1$=CHR$(200)+CHR$(201)+CHR$(20 [2566]
2)+CHR$(203)
980 c2$=CHR$(202)+CHR$(203)+CHR$(20 [3264]
0)+CHR$(201)
990 c3$=CHR$(204)+CHR$(205)+CHR$(20 [3962]
6)+CHR$(207)
1000 c4$=CHR$(208)+CHR$(209)+CHR$(2 [2226]
10)+CHR$(211)
1010 c5$=CHR$(212)+CHR$(10)+CHR$(8) [3890]
+CHR$(202)+CHR$(203)+CHR$(8)+CHR$(2
13)
1020 c6$=CHR$(214)+CHR$(215)         [1697]
1030 c7$=CHR$(216)+CHR$(217)         [2092]
1040 mart$=CHR$(218)+CHR$(219)       [2171]
1050 tr$=CHR$(22)+CHR$(1)            [806]
1060 nr$=CHR$(22)+CHR$(0)            [700]
1070 DEF FN po(x,y)=&C000+(y-1)*80+ [585]
(x-1)*2
1080 sp(1)=&90C3:sp(2)=&9104:sp(3)= [2616]
&9041:sp(4)=&9000:sens=1
1090 WINDOW #1,1,40,1,21:WINDOW #2, [2362]
15,25,20,22
1100 sc=0:rec=0:fin=0:vie=5          [911]
1110 ta=INT(RND*19)+1:vit1=50:vit2= [1927]
550
1120 REM :::::::::::::::::::::::::   [1823]
```

```
1130 REM :                        :   [419]
1140 REM :     TAB DES SCORES     :   [889]
1150 REM :                        :   [419]
1160 REM ::::::::::::::::::::::::::   [1823]
1170 PEN 2:LOCATE 4,23:PRINT "SC      [4738]
              VI    TA":LOCATE
4,24:PRINT "HI
 TIME"
1180 GOSUB 2480:GOSUB 2520:GOSUB 25 [2990]
60:GOSUB 2600:GOSUB 2620
1190 PLOT 44,12,1:DRAW 44,52:DRAW 1 [2884]
96,52:DRAW 196,12:DRAW 44,12
1200 PLOT 428,12:DRAW 428,52:DRAW 5 [2531]
96,52:DRAW 596,12:DRAW 428,12
1210 FOR H=12 TO 36:PLOT 208,H:DRAW [2812]
 416,H:NEXT
1220 PEN 3:LOCATE 15,24:PRINT "SUPE [1909]
R GLUPS"
1230 REM ::::::::::::::::::::::::::   [1823]
1240 REM :                        :   [419]
1250 REM :       TABLEAU +1       :   [574]
1260 REM :                        :   [419]
1270 REM ::::::::::::::::::::::::::   [1823]
1280 DIM jeu(12,12):DIM a1$(11),b1$ [2578]
(11),d1$(11)
1290 CLS #1:CLS #2:GOSUB 2600       [1896]
1300 IF ta>20 THEN ta=1             [806]
1310 ON ta GOTO 1320,1330,1340,1350 [3686]
,1360,1370,1380,1390,1400,1410,1420
,1430,1440,1450,1460,1470,1480,1490
,1500,1510
1320 RESTORE 3330:GOTO 1520         [695]
1330 RESTORE 3350:GOTO 1520         [1080]
1340 RESTORE 3370:GOTO 1520         [1712]
1350 RESTORE 3390:GOTO 1520         [1326]
1360 RESTORE 3410:GOTO 1520         [1094]
1370 RESTORE 3430:GOTO 1520         [1326]
1380 RESTORE 3450:GOTO 1520         [695]
1390 RESTORE 3470:GOTO 1520         [856]
1400 RESTORE 3490:GOTO 1520         [2032]
1410 RESTORE 3510:GOTO 1520         [1080]
1420 RESTORE 3530:GOTO 1520         [1094]
1430 RESTORE 3550:GOTO 1520         [968]
1440 RESTORE 3570:GOTO 1520         [695]
1450 RESTORE 3590:GOTO 1520         [1118]
1460 RESTORE 3610:GOTO 1520         [1792]
1470 RESTORE 3630:GOTO 1520         [1080]
1480 RESTORE 3650:GOTO 1520         [1712]
1490 RESTORE 3670:GOTO 1520         [968]
1500 RESTORE 3690:GOTO 1520         [1568]
1510 RESTORE 3710:GOTO 1520         [1326]
1520 vit1=vit1-2:vit2=vit2-10:GOSUB [3061]
 1970:GOSUB 2560
1530 REM ::::::::::::::::::::::::::   [1823]
1540 REM :                        :   [419]
1550 REM :  ROUTINE PRINCIPALE    :   [2225]
1560 REM :                        :   [419]
1570 REM ::::::::::::::::::::::::::   [1823]
1580 a=28:b=15:a1=a:b1=b:jx=6:jy=11 [3194]
:mx=10:my=6:mx1=mx:my1=mY:jx1=6:jy1
=2
1590 CALL &A060,FN po(a,b):sens=1   [1744]
1600 CALL &A160,FN po(mx,my)        [1831]
1610 CALL &A000,FN po(a,b),sp(1)    [1333]
1620 CALL &A000,FN po(mx,my),&9082  [1114]
1630 EVERY vit1,2 GOSUB 2310:EVERY  [2519]
vit2,3 GOSUB 3070
1640 IF fin=1 THEN 2680             [1009]
1650 IF INKEY(ht)=0 THEN 1730       [1272]
1660 IF INKEY(da)=0 THEN sens=sens+ [2152]
1:GOTO 1690
1670 IF INKEY(ga)=0 THEN sens=sens- [1870]
1:GOTO 1700
1680 GOTO 1640                      [307]
1690 IF sens>4 THEN sens=1:GOTO 171 [1001]
0
1700 IF sens<1 THEN sens=4          [588]
1710 CALL &A080,FN po(a1,b1):CALL & [3201]
A000,FN po(a,b),sp(sens)
1720 FOR t=1 TO 15:SOUND 1,50-t,1,5 [4093]
:NEXT t:GOTO 1640
1730 ON sens GOTO 1740,1770,1750,17 [1253]
60
1740 jy=jy-1:IF jeu(jx,jy)>0 THEN 1 [2838]
780 ELSE jy=jy+1:GOTO 1640
1750 jy=jy+1:IF jeu(jx,jy)>0 THEN 1 [3662]
780 ELSE jy=jy-1:GOTO 1640
1760 jx=jx-1:IF jeu(jx,jy)>0 THEN 1 [2210]
780 ELSE jx=jx+1:GOTO 1640
1770 jx=jx+1:IF jeu(jx,jy)>0 THEN 1 [3077]
780 ELSE jx=jx-1:GOTO 1640
1780 a=(jy*2)+(jx*2)-6:b=(10+jy)-jx [1570]
1790 IF a=mx AND b=my THEN CALL &A0 [3570]
80,FN po(a1,b1):fin=1:mu=REMAIN(2):
GOTO 1640
1800 CALL &A080,FN po(a1,b1)        [1457]
1810 CALL &A060,FN po(a,b):CALL &A0 [2109]
00,FN po(a,b),sp(sens)
1820 SOUND 1,100,2,5                [1293]
1830 IF TEST(a1*16,378-(b1*16))=3 T [2298]
HEN GOSUB 1860
1840 a1=a:b1=b:IF jeu(jx,jy)=10 THE [1637]
N 1890
1850 GOTO 1640                      [307]
1860 DI:PEN 2:LOCATE a1,b1+1:PRINT  [5823]
c6$:LOCATE a1,b1+2:PRINT c7$:EI:SOU
ND 2,300,2,6
1870 pic=pic+1:IF pic=nbpic THEN GO [3907]
SUB 3170
1880 sc=sc+100:RETURN               [1073]
1890 mu=REMAIN(2):tp=REMAIN(3)      [1124]
1900 GOSUB 2480:FOR t=1 TO 200:SOUN [5549]
D 1,t,1,14:NEXT t:ta=ta+1:pic=0:cen
t=cent+4.5
1910 ERASE jeu:ERASE a1$,b1$,d1$:GO [1585]
TO 1280
1920 REM ::::::::::::::::::::::::::   [1823]
1930 REM :                        :   [419]
1940 REM :   DESSIN TABLEAUX      :   [1979]
1950 REM :                        :   [419]
1960 REM ::::::::::::::::::::::::::   [1823]
1970 FOR h=1 TO 11:READ a1          [1407]
1980 a1$(h)=BIN$(a1,11)             [698]
1990 NEXT                           [350]
2000 FOR h=1 TO 11:FOR g=1 TO 11    [1895]
2010 b1$=MID$(a1$(h),g,1):b1=VAL(b1 [2145]
$)
2020 IF b1=1 THEN 2050              [316]
2030 NEXT g,h                       [389]
2040 GOTO 2150                      [349]
2050 ax=(g*2)+(h*2)-5:by=(12+h)-g   [1262]
2060 PEN 1:LOCATE ax,by:PRINT tr$;c [5610]
1$:LOCATE ax,by+1:PRINT c2$:PEN 2:L
OCATE ax,by:PRINT c3$:LOCATE ax,by+
1:PRINT c4$
2070 jeu(g,h+1)=1                   [572]
2080 b2$=MID$(a1$(h),g-1,1):b2=VAL( [1493]
b2$)
2090 IF b2=1 THEN 2110              [532]
2100 PEN 1:LOCATE ax,by+1:PRINT c5$ [1165]
```

```
2110 d1$=MID$(a1$(h+1),g,1):d1=VAL( [1427]
d1$)
2120 IF d1=1 THEN 2030                [544]
2130 PEN 1:LOCATE ax+2,by+2:PRINT C [8721]
HR$(212)+CHR$(201):LOCATE ax+2,by+2
:PRINT CHR$(200):LOCATE ax+3,by+1:P
RINT CHR$(213)
2140 GOTO 2030                        [365]
2150 REM :::: DESSIN COURONNES :::: [1439]
2160 FOR h=1 TO 11:READ a1          [1407]
2170 a1$(h)=BIN$(a1,11)               [698]
2180 NEXT                             [350]
2190 FOR h=1 TO 11:FOR g=1 TO 11    [1895]
2200 b1$=MID$(a1$(h),g,1):b1=VAL(b1 [2145]
$)
2210 IF b1=1 THEN 2230                [312]
2220 NEXT g,h:READ nbpic,sort:RETUR [2039]
N
2230 ax=(g*2)+(h*2)-4:by=(12+h)-g   [1834]
2240 PEN 1:LOCATE ax,by:PRINT c6$:P [4400]
EN 3:LOCATE ax,by+1:PRINT c7$
2250 jeu(g,h+1)=2:GOTO 2220           [735]
2260 REM ::::::::::::::::::::::::   [1823]
2270 REM :                      :     [419]
2280 REM : DEPLACEMENT BIG BLUE :   [1727]
2290 REM :                      :     [419]
2300 REM ::::::::::::::::::::::::   [1823]
2310 DI                                [84]
2320 IF jx>jx1 AND jeu(jx1+1,jy1)>0 [2721]
THEN jx1=jx1+1:GOTO 2360
2330 IF jy<jy1 AND jeu(jx1,jy1-1)>0 [3501]
THEN jy1=jy1-1:GOTO 2360
2340 IF jy>jy1 AND jeu(jx1,jy1+1)>0 [2611]
THEN jy1=jy1+1:GOTO 2360
2350 IF jx<jx1 AND jeu(jx1-1,jy1)>0 [3020]
THEN jx1=jx1-1
2360 mx=(jy1*2)+(jx1*2)-6:my=(10+jY [2372]
1)-jX1
2370 IF mx=a AND my=b THEN mu=REMAI [2095]
N(2):GOTO 2420
2380 CALL &A180,FN po(mx1,my1):CALL [4565]
&A160,FN po(mx,my):CALL &A000,FN p
o(mx,my),&9082
2390 mx1=mx:my1=my:                   [782]
2400 EI:RETURN                        [416]
2410 REM un os ? TEL 96 38 94 24 (d [2390]
emandez CLAUDE)
2420 EI:CALL &A180,FN po(mx1,my1):d [2326]
er=1:fin=1:RETURN
2430 REM ::::::::::::::::::::::::   [1823]
2440 REM :                      :     [419]
2450 REM :  GESTION DES SCORES  :   [1780]
2460 REM :                      :     [419]
2470 REM ::::::::::::::::::::::::   [1823]
2480 PEN 1:IF sc=0 THEN LOCATE 7,23 [2902]
:PRINT nr$;"000000":RETURN
2490 IF sc<1000 THEN LOCATE 10,23:P [5186]
RINT nr$;"   ":LOCATE 9,23:PRINT tr
$;sc:RETURN
2500 IF sc<10000 THEN LOCATE 9,23:P [5244]
RINT nr$;"    ":LOCATE 8,23:PRINT t
r$;sc:RETURN
2510 IF sc>10000 THEN LOCATE 8,23:P [5690]
RINT nr$;"     ":LOCATE 7,23:PRINT
tr$;sc:RETURN
2520 PEN 1:LOCATE 7,24:PRINT nr$;"0 [2504]
00000":IF rec=0 THEN RETURN
2530 IF rec<1000 THEN LOCATE 10,24: [4864]
PRINT nr$;"   ":LOCATE 9,24:PRINT t
r$;rec:RETURN
2540 IF rec<10000 THEN LOCATE 9,24: [4998]
PRINT nr$;"    ":LOCATE 8,24:PRINT
tr$;rec:RETURN
2550 IF rec>10000 THEN LOCATE 8,24: [4979]
PRINT nr$;"     ":LOCATE 7,24:PRINT
tr$;rec:RETURN
2560 PEN 0:LOCATE 36,23:PRINT CHR$( [2915]
143)+CHR$(143)
2570 IF ta<10 THEN 2580 ELSE 2590   [1722]
2580 PEN 1:LOCATE 36,23:PRINT"0":LO [3498]
CATE 36,23:PRINT tr$;ta:RETURN
2590 PEN 1:LOCATE 35,23:PRINT tr$;t [2298]
a:RETURN
2600 PEN 3:LOCATE 32,24:PRINT tr$;C [5702]
HR$(222)+CHR$(222):PEN 1:LOCATE 34,
24:PRINT CHR$(222)+CHR$(222)+CHR$(2
22)+CHR$(222)
2610 temps=12:RETURN                [1111]
2620 PEN 0:LOCATE 31,23:PRINT CHR$( [4220]
143):PEN 1:LOCATE 30,23:PRINT vie:R
ETURN
2630 REM ::::::::::::::::::::::::   [1823]
2640 REM :                      :     [419]
2650 REM :        VIE -1        :     [647]
2660 REM :                      :     [419]
2670 REM ::::::::::::::::::::::::   [1823]
2680 FOR h=1 TO 10:OUT &BC00,2:OUT  [7666]
&BD00,49:FOR t=1 TO 70:NEXT:SOUND 4
,1500+h*100,20,7,0,0,h:OUT &BC00,2:
OUT &BD00,43:FOR t=1 TO 70:NEXT:NEX
T h:OUT &BC00,2:OUT &BD00,46
2690 FOR t=1 TO 2000:NEXT t         [1676]
2700 IF der=1 THEN CALL &A080,FN po [4063]
(a1,b1) ELSE CALL &A180,FN po(mx1,m
y1)
2710 der=0:fin=0:vie=vie-1:GOSUB 24 [2643]
80:GOSUB 2620
2720 CALL &A180,FN po(mx1,my1):CALL [3852]
&A080,FN po(a1,b1)
2730 IF vie=0 THEN GOSUB 2800         [813]
2740 GOSUB 2480:GOSUB 2600:tp=REMAI [2741]
N(3):GOTO 1580
2750 REM ::::::::::::::::::::::::   [1823]
2760 REM :                      :     [419]
2770 REM :        PERDU         :     [934]
2780 REM :                      :     [419]
2790 REM ::::::::::::::::::::::::   [1823]
2800 ENT 4,20,3,5:SOUND 4,20,100,15 [3258]
,0,4:FOR t=1 TO 2000:NEXT
2810 CLS #1:CLS #2:tp=REMAIN(3):mu= [2346]
REMAIN(2)
2820 IF sc>rec THEN rec=sc:GOSUB 25   [767]
20
2830 RESTORE 2860:FOR h=1 TO 9:READ [4332]
ax,by:GOSUB 3230:NEXT
2840 FOR h=1 TO 3:READ ax,by:GOSUB  [2840]
3250:NEXT
2850 FOR h=1 TO 3:READ ax,by:GOSUB  [2139]
3260:NEXT
2860 DATA 15,8,19,8,23,8,17,7,21,7, [3818]
19,6,17,9,21,9,19,10,15,8,17,9,19,1
0,19,10,21,9,23,8
2870 CALL &A000,FN po(19.5,5),sp(4) [1394]
2880 WHILE INKEY$<>"":WEND          [1786]
2890 LOCATE 14,14:PEN 3:PRINT "REVI [3040]
TALISATION"
2900 LOCATE 5,16:PEN 2:PRINT "VOUS  [4226]
AVEZ FAIT ";CENT;"% DU PARCOURS"
2910 cent=0                           [352]
2920 LOCATE 5,18:PEN 1:PRINT "POUR  [4666]
```

SUPER GLUPS 3D

```
REJOUER APPUYEZ SUR <ENTER>"
2930 PEN 0:LOCATE 20,4:PRINT CHR$(1  [5560]
43)+CHR$(143):LOCATE 20,3:PEN 1:PRI
NT MART$
2940 FOR t=1 TO 200:NEXT             [1050]
2950 PEN 0:LOCATE 20,3:PRINT CHR$(1  [5185]
43)+CHR$(143):LOCATE 20,4:PEN 1:PRI
NT MART$;
2960 SOUND 2,50,5,15:FOR t=1 TO 200  [3387]
:NEXT
2970 IF INKEY$=CHR$(13) THEN 2990    [1069]
2980 GOTO 2930                       [337]
2990 VIE=5:GOSUB 2620:SC=0:GOSUB 24  [1995]
80:PIC=0
3000 ta=INT(RND*19)+1:GOSUB 2600     [935]
3010 vit1=50:vit2=550:GOTO 1910      [1255]
3020 REM ::::::::::::::::::::::::    [1823]
3030 REM :                      :    [419]
3040 REM :       TEMPS -1       :    [742]
3050 REM :                      :    [419]
3060 REM ::::::::::::::::::::::::    [1823]
3070 DI:LOCATE 2,2:PRINT CHR$(7):te  [4808]
mps=temps-1:IF INT(temps/2)=temps/2
 THEN GOTO 3090 ELSE 3080
3080 PEN 0:LOCATE 32+((temps-1)/2),  [3815]
24:PRINT tr$;CHR$(138):GOTO 3110
3090 PEN 0:LOCATE 32+(temps/2),24:P  [2705]
RINT tr$;CHR$(133)
3100 IF temps=0 THEN fin=1:tp=REMAI  [2452]
N(3):mu=REMAIN(2)
3110 EI:RETURN                       [416]
3120 REM ::::::::::::::::::::::::    [1823]
3130 REM :                      :    [419]
3140 REM :   AFFICHAGE SORTIE   :    [1643]
3150 REM :                      :    [419]
3160 REM ::::::::::::::::::::::::    [1823]
3170 FOR h=1 TO 30:SOUND 1,100-h,2,  [2133]
5:NEXT
3180 ON sort GOTO 3190,3200,3210,32  [1738]
20
3190 ax=9:by=5:GOSUB 3230:GOSUB 325  [4678]
0:by=7:GOSUB 3230:jeu(7,1)=10:RETUR
N
3200 ax=31:by=6:GOSUB 3230:GOSUB 32  [4177]
60:by=8:GOSUB 3230:jeu(12,7)=10:RET
URN
3210 ax=31:by=16:GOSUB 3230:GOSUB 3  [2945]
250:GOSUB 3260:jeu(7,12)=10:RETURN

3220 ax=9:by=17:GOSUB 3230:GOSUB 32  [3316]
50:GOSUB 3260:jeu(1,7)=10:RETURN
3230 PEN 1:LOCATE ax,by:PRINT tr$;c  [5610]
1$:LOCATE ax,by+1:PRINT c2$:PEN 2:L
OCATE ax,by:PRINT c3$:LOCATE ax,by+
1:PRINT c4$
3240 RETURN                          [555]
3250 PEN 1:LOCATE ax,by+1:PRINT c5$  [1828]
:RETURN
3260 PEN 1:LOCATE ax+2,by+2:PRINT C  [8721]
HR$(212)+CHR$(201):LOCATE ax+2,by+2
:PRINT CHR$(200):LOCATE ax+3,by+1:P
RINT CHR$(213)
3270 RETURN                          [555]
3280 REM ::::::::::::::::::::::::    [1823]
3290 REM :                      :    [419]
3300 REM :   DATAS DES 20 TAB   :    [1096]
3310 REM :                      :    [419]
3320 REM ::::::::::::::::::::::::    [1823]
3330 DATA 1023,1023,1023,1023,1023,  [2109]
1023,1023,1023,1023,1023,0
3340 DATA 1,3,5,148,254,200,224,54,  [1620]
0,0,0,25,4
3350 DATA 1023,529,529,529,529,1023  [2862]
,529,529,529,1023,0
3360 DATA 529,0,0,0,529,0,0,0,0,529  [1479]
,0,9,3
3370 DATA 1023,513,1023,16,16,16,10  [2044]
23,581,581,1015,0
3380 DATA 528,0,1,0,16,0,512,5,64,5  [1356]
13,0,10,1
3390 DATA 1023,545,765,645,701,693,  [2692]
645,765,529,1023,0
3400 DATA 513,32,128,0,48,48,0,4,0,  [1567]
513,0,11,2
3410 DATA 1023,1023,1023,1023,1023,  [2109]
1023,1023,1023,1023,1023,0
3420 DATA 0,256,128,64,32,16,8,4,2,  [1539]
0,0,8,1
3430 DATA 1023,1023,1023,1023,1023,  [2109]
1023,1023,1023,1023,1023,0
3440 DATA 256,8,128,1,32,512,64,2,1  [1347]
6,4,0,10,3
3450 DATA 1023,853,513,1023,513,102  [2677]
3,513,1023,513,1023,0
3460 DATA 0,340,0,32,1,32,0,544,0,3  [1682]
3,0,11,2
3470 DATA 1023,633,585,975,633,585,  [2376]
975,633,585,1023,0
3480 DATA 512,65,0,256,528,0,129,32  [1840]
,512,2,0,11,4
3490 DATA 1023,1023,1023,1023,1023,  [2109]
1023,1023,1023,1023,1023,0
3500 DATA 510,510,510,510,510,510,5  [2469]
10,510,510,510,0,80,1
3510 DATA 1023,951,879,765,1019,983  [2414]
,957,891,1015,1023,0
3520 DATA 34,304,8,65,512,2,8,64,26  [1569]
0,0,0,14,2
3530 DATA 943,681,683,681,763,681,6  [1874]
83,681,681,943,0
3540 DATA 681,0,651,0,82,0,651,0,0,  [1373]
681,0,23,4
3550 DATA 943,697,683,749,697,687,6  [2292]
72,703,703,959,0
3560 DATA 676,49,130,580,9,2,544,12  [1850]
8,0,640,0,20,2
3570 DATA 943,943,943,1023,943,943,  [1774]
943,943,32,1023,0
3580 DATA 681,0,258,0,640,260,160,5  [2199]
21,0,513,0,18,4
3590 DATA 1023,545,687,761,1021,633  [1914]
,765,597,529,1023,0
3600 DATA 545,0,170,128,260,0,132,8  [1888]
4,0,545,0,18,1
3610 DATA 693,1023,510,1023,510,102  [2723]
3,510,1023,510,171,0
3620 DATA 693,769,0,513,0,513,0,513  [3445]
,256,171,0,21,3
3630 DATA 1023,683,545,883,553,1023  [2642]
,617,803,681,1023,0
3640 DATA 0,170,0,338,8,258,72,258,  [1888]
168,0,0,18,4
3650 DATA 1023,1023,1023,1023,1023,  [2109]
1023,1023,1023,1023,1023,0
3660 DATA 273,283,277,273,465,0,56,  [2049]
32,32,56,0,28,1
3670 DATA 495,231,627,825,1023,1023  [2454]
,926,828,632,240,0
3680 DATA 264,132,578,289,192,0,130  [2296]
,260,520,16,0,19,3
```

```
3690 DATA 1023,513,1021,517,757,693 [2832]
,645,765,513,1023,0
3700 DATA 513,0,0,0,48,48,0,0,0,513 [1204]
,0,8,2
3710 DATA 1023,1023,1023,1023,1023, [2109]
1023,1023,1023,1023,1023,0
3720 DATA 0,1022,1008,896,1020,1022 [2469]
,960,768,960,0,0,45,1
3730 REM :::::::::::::::::::::::::: [1924]
3740 REM Vous pouvez creer vos prop [2631]
res tableaux.
3750 REM Dans les datas, chaque tab [5675]
 occupe deux lignes.les tableaux so
nt fait dans une matrice de 10 X 10
3760 REM La numerotation est identi [3820]
que  a celle utilisee pour redefini
r un caractere.
3770 REM 1ere ligne (11 nombres pou [8943]
r la plate forme),addition des dix
lignes de la matrice plus un zero p
our l'algorithme du systeme
3780 REM 2eme ligne (13 nombres)add [6764]
ition des dix lignes de la matrice
pour les couronnes  cette fois ci

3790 REM un zero pour l'algorithme  [5213]
du systeme puis le nb de couronnes
et le chiffre de la sortie (entre 1
 et 4)
```

ADDITIF TECHNIQUE

Variables de base

C1$ à C7$: *CHR$ redéfinis permettant la création du damier*

Mart$: *CHR$ redéfinis pour le marteau*

Tr$: *active mode transparent*

Nr$: *désactive mode transparent*

DEF FN po(x,y) : *transforme un locate x, y en adresse écran*

Sp(1) à Sp(4) : *add mémoire des sprites du glups*
Autres variables (vie, fin, sc, ta...) : sans commentaire

Algorithme du dessin des tableaux

Le dessin de chaque tableau est tracé dans une matrice de 10 x 10. Comme pour un CHR$, une numérotation est réalisée mais sur 10 lignes et 10 colonnes au lieu de 8. Ces 10 nombres sont rangés en datas à la fin du programme. Les explications sont incluses dans les REM.

Les routines en langage machine

Le déplacement des sprites dans un décor sans provoquer d'effacement, nécessite la technique des masques. C'est-à-dire :

A- Sauver en mémoire la portion de décor sur laquelle sera affichée le sprite.
B- Afficher le sprite en AND,
C- Test de déplacement ? Si oui...
D- Effacement du sprite par restitution du morceau de décor préalablement sauvegardé.
E- Cacul de la nouvelle position du sprite et GOTO en A.

Pour ce faire, les 3 routines suivantes sont utilisées :

CALL &A000,X,Y : *affiche en AND à l'adresse écran X, un sprite de 4 octets sur 16 lignes de pixels, se trouvant en mémoire à l'adresse Y.*

CALL &A060,X : *sauvegarde en &A400, une portion de l'écran de 4 octets sur 16 lignes, X étant l'adresse de l'octet le plus haut et le plus à gauche de cette matrice.*

CALL &A080,X : *restitue en X, ce qui avait été précédemment sauvegardé.*

```
PAGE 1                              ;AFFICHE SPRITE :
                                    ;CALL &A000, Add écran, Add mémoire
A000 DD5E00         LD   E,(IX+0)   ;Add du sprite
A003 DD5601         LD   D,(IX+1)   ;dans DE
A006 DD6E02         LD   L,(IX+2)   ;Add écran
A009 DD6603         LD   H,(IX+3)   ;dans HL
A00C 0610           LD   B,10H      ;16 lignes de pixels
A00E CD19BD         CALL 0BD19H     ;Retour rayon
A011 F3             DI
A012 C5     LA012:  PUSH BC
A013 E5             PUSH HL
A014 0604           LD   B,4        ;4 octets par ligne
A016 1A     LA016:  LD   A,(DE)     ;4 pixels par octets
A017 E688           AND  88H        ;
A019 2806           JR   Z,LA021    ;Le pixel du sprite n'est
A01B 4F             LD   C,A        ;affiché que s'il est différent
A01C 7E             LD   A,(HL)     ;de zéro. Sinon, le sprite se
A01D E677           AND  'w'        ;retrouverait au milieu d'un
A01F B1             OR   C          ;carré noir.
A020 77             LD   (HL),A     ;
A021 1A     LA021:  LD   A,(DE)     ;\
A022 E644           AND  'D'        ;:
A024 2806           JR   Z,LA02C    ;:
A026 4F             LD   C,A        ; > Pixel 2
A027 7E             LD   A,(HL)     ;:
A028 E6BB           AND  0BBH       ;:
A02A B1             OR   C          ;:
A02B 77             LD   (HL),A     ;/
A02C 1A     LA02C:  LD   A,(DE)     ;\
A02D E622           AND  22H        ;:
A02F 2806           JR   Z,LA037    ;:
A031 4F             LD   C,A        ; > Pixel 3
A032 7E             LD   A,(HL)     ;:
A033 E6DD           AND  0DDH       ;:
A035 B1             OR   C          ;:
A036 77             LD   (HL),A     ;/
A037 1A     LA037:  LD   A,(DE)     ;\
A038 E611           AND  11H        ;:
A03A 2806           JR   Z,LA042    ;:
A03C 4F             LD   C,A        ; > Pixel 4
A03D 7E             LD   A,(HL)     ;:
A03E E6EE           AND  0EEH       ;:
A040 B1             OR   C          ;:
A041 77             LD   (HL),A     ;/
A042 23     LA042:  INC  HL         ;
A043 13             INC  DE         ;
A044 10D0           DJNZ LA016      ;
A046 E1             POP  HL         ;
A047 CD51A0         CALL LA051      ;calcul de l'add de
A04A C1             POP  BC         ;la ligne inférieure
A04B 10C5           DJNZ LA012      ;
A04D FB             EI              ;
A04E C9             RET             ;
A04F 00             NOP
A050 00             NOP
A051 7C     LA051:  LD   A,H        ;\
A052 C608           ADD  A,8        ; >Add écran + 1 ligne
A054 67             LD   H,A        ;/
A055 D0             RET  NC
A056 D5             PUSH DE
A057 1150C0         LD   DE,0C050H
A05A 19             ADD  HL,DE

PAGE 2

A05B D1             POP  DE
A05C C9             RET

PAGE 1

A060 1100A4         LD   DE,0A400H;CALL &A060,X : sauvegarde
A063 DD6E00         LD   L,(IX+0) ;d'une partie de l'écran
A066 DD6601         LD   H,(IX+1) ;
A069 0610           LD   B,10H    ;
A06B C5     LA06B:  PUSH BC       ;
A06C E5             PUSH HL       ;
A06D 0604           LD   B,4      ;
A06F 7E     LA06F:  LD   A,(HL)   ;
A070 12             LD   (DE),A   ;
A071 13             INC  DE       ;
A072 23             INC  HL       ;
A073 10FA           DJNZ LA06F    ;
A075 E1             POP  HL       ;
A076 CD51A0         CALL 0A051H   ;
A079 C1             POP  BC       ;
A07A 10EF           DJNZ LA06B    ;
A07C C9             RET           ;

PAGE 1

A080 1100A4         LD   DE,0A400H;CALL &A080,X : restitution
A083 DD6E00         LD   L,(IX+0) ;d'une partie de l'écran
A086 DD6601         LD   H,(IX+1) ;
A089 0610           LD   B,10H    ;
A08B C5     LA08B:  PUSH BC       ;
A08C E5             PUSH HL       ;
A08D 0604           LD   B,4      ;
A08F 1A     LA08F:  LD   A,(DE)   ;
A090 77             LD   (HL),A   ;
A091 23             INC  HL       ;
A092 13             INC  DE       ;
A093 10FA           DJNZ LA08F    ;
A095 E1             POP  HL       ;
A096 CD51A0         CALL 0A051H   ;
A099 C1             POP  BC       ;
A09A 10EF           DJNZ LA08B    ;
A09C C9             RET           ;
```

HARDCOPY et FENETRE (6128)

L'utilitaire HARDCOPY permet l'impression d'images écran (DMP 2000 et compatible EPSON), en respectant les nuances spécifiques au mode utilisé : 16 en mode 0, 4 en mode 1, et 2 en mode 2 (voir exemple joint).

Entrez les codes hexadécimaux du premier listing à l'aide du programme AMSAISIE (reportez-vous à son mode d'emploi). Spécifiez A000 comme adresse de début et sauvez le langage machine par l'option « S » sous le nom **HARDA06C**. Si vous ne souhaitez pas saisir en une seule fois la totalité des codes, morcelez votre travail en créant plusieurs fichiers **(HARD1, HARD2**, etc). Ces derniers devront ultérieurement être chargés à la suite après un **MEMORY &9FFF** et sauvegardés ainsi dans un fichier unique : **SAVE « HARDA06C .BIN », b, &A000,&56B.**

L'utilisation nécessite un **MEMORY &9FFF : LOAD" "HARDA06C.BIN"** et un **CALL &A06C** à la fin du programme de chargement ou de création d'image (ne pas oublier de mettre préalablement l'imprimante en fonction et de préciser le mode dans votre programme).

Un exemple d'utilisation vous est offert par le second listing (dans le cas d'une utilisation K7, veillez à sauvegarder celui-ci « avant » la routine binaire).

L'utilitaire FENETRE (troisième listing), qui permet le calibrage d'une fenêtre écran destinée à l'impression (chargement de **HARDA06C.BIN** inclu en début de programme de chargement ou de création d'image.



Les contours du cadre qui se superpose à l'affichage, sont modifiables par la manette de jeu (peu pratique) ou les touches directionnelles (recommandées).

— Chaque direction (haut, bas, gauche ou droite) déplace la limite correspondante du cadre dans le sens spécifié. Les déplacements inverses s'effectuent par appui simultané sur SHIFT/Direction (ou FIRE/Direction).

— COPY permet la sortie avec impression.

— ENTER ou RETURN autorise la sortie sans impression. Le clignotement, en action pendant la définition des limites, cesse dès la sortie.

Philippe Canouil

Nota : concernant la sortie sans impression, il est possible de sauvegarder comme suit le programme d'édition avec la fenêtre définie :

```
SAVE"FENEA06C.BIN",b,&A000,&58C
```

```
A000:96 A2 9B A2 A0 A2 A5 A2:9E
A008:AA A2 AF A2 B4 A2 B9 A2:F6
A010:BE A2 C3 A2 C8 A2 CD A2:4E
A018:D2 A2 D7 A2 DC A2 DF A2:A4
A020:CD A1 D2 A1 D7 A1 DA A1:94
A028:2A A5 03 A5 DC A4 B5 A4:18
A030:8E A4 67 A4 40 A4 19 A4:AE
A038:F2 A3 CB A3 A4 A3 7D A3:42
A040:56 A3 2F A3 08 A3 E1 A2:D9
A048:1B A2 06 A2 F1 A1 DC A1:5C
A050:0B 13 1B 03 19 02 0A 1A:6B
A058:12 1F 0F 17 07 0D 0E 00:71
A060:06 05 1D 15 1E 16 0C 04:81
A068:18 1C 14 00 3E 00 32 6B:2B
```

```
A070:A0 FD 21 28 A0 DD 21 50:E4
A078:A0 0E 10 21 E6 B7 06 10:AA
A080:7E DD BE 00 28 0D 23 10:A1
A088:F7 3E 00 B1 28 59 DD 23:8F
A090:C3 7B A0 E5 21 00 A0 78:2C
A098:2F C6 11 07 6F 22 A2 A0:18
A0A0:ED 5B 11 A1 FD 22 A9 A0:A2
A0A8:21 00 A0 C5 ED A0 ED A0:E8
A0B0:C1 FD 23 FD 23 78 D6 0D:AC
A0B8:FA E2 A0 21 20 A0 78 2F:5C
A0C0:C6 11 07 85 6F 22 CA A0:BE
A0C8:ED 5B 00 A0 21 48 A0 3A:93
A0D0:6B A0 07 85 6F 3A 6B A0:BB
A0D8:3C 32 6B A0 C5 ED A0 ED:30
A0E0:A0 C1 0D E1 C3 86 A0 00:B8
A0E8:CD BA BB CD E7 BB 32 B0:1B
A0F0:A1 CD 5C A1 21 8F 01 22:CE
A0F8:B1 A1 11 00 00 3E 06 32:71
A100:B3 A1 CD 6C A1 3A B3 A1:5D
A108:47 CD 11 BC DA 8A A2 CA:5A
A110:C1 A1 0E 00 E5 D5 C5 CD:6D
A118:F0 BB C1 D1 21 B0 A1 BE:26
A120:E1 37 20 01 A7 CB 11 2B:A8
A128:10 EA CD A4 A1 79 CD 9B:B6
A130:A1 13 E5 21 7C 02 37 ED:2D
A138:52 E1 38 05 2A B1 A1 18:DD
A140:C4 23 7C B5 C8 2B 11 00:FD
A148:00 22 B1 A1 3E 03 BD 20:7B
A150:B1 7C B4 20 AD 3E 04 32:13
A158:B3 A1 18 A6 3E 1B CD 9B:CC
A160:A1 3E 33 CD 9B A1 3E 12:6C
A168:CD 9B A1 C9 E5 3E 42 CD:0D
A170:1E BB E1 28 02 E1 C9 3E:DD
A178:0D CD 9B A1 3E 0A CD 9B:DF
A180:A1 3E 1B CD 9B A1 3E 2A:8C
A188:CD 9B A1 3E 04 CD 9B A1:7D
A190:3E 7C CD 9B A1 3E 02 CD:01
A198:9B A1 C9 CD 2E BD 38 FB:29
A1A0:CD 2B BD C9 3A B3 A1 FE:4B
A1A8:06 C8 AF CB 11 CB 11 C9:47
A1B0:00 00 00 00 00 00 00 00:51
A1B8:00 2B 2B 05 10 03 C3 30:BA
A1C0:A2 E5 D5 C5 CD F0 BB C1:BB
A1C8:D1 FE 00 E1 CA 1B A2 D6:76
A1D0:01 CA 06 A2 D6 01 CA F1:76
A1D8:A1 C3 DC A1 3A B5 A1 CD:B7
A1E0:5C A2 32 B5 A1 3A B6 A1:98
A1E8:CD 5C A2 32 B6 A1 C3 B9:59
A1F0:A1 3A B5 A1 CD 4A A2 32:AD
A1F8:B5 A1 3A B6 A1 CD 53 A2:42
A200:32 B6 A1 C3 B9 A1 3A B5:37
A208:A1 CD 4A A2 32 B5 A1 3A:C6
A210:B6 A1 CD 42 A2 32 B6 A1:43
A218:C3 B9 A1 3A B5 A1 CD 42:76
A220:A2 32 B5 A1 3A B6 A1 CD:4A
A228:42 A2 32 B6 A1 C3 B9 A1:54
A230:CD 51 A5 CD 5D A5 3E 00:A2
A238:32 B5 A1 32 B6 A1 13 C3:C1
A240:31 A1 4F B7 CB 11 CB 11:72
A248:79 C9 4F 37 CB 11 B7 CB:10
A250:11 79 C9 4F B7 CB 11 37:5E
A258:CB 11 79 C9 4F 37 CB 11:7A
A260:37 CB 11 79 C9 2B 2B 05:B2
A268:10 20 CD 51 A5 CD 5D A5:CC
A270:CD 62 A5 CD 67 A5 3E 00:FD
A278:32 B5 A1 32 B6 A1 32 B7:14
A280:A1 32 B8 A1 13 13 13 C3:4A
A288:31 A1 E5 D5 C5 CD F0 BB:F3
A290:C1 D1 FE 00 E1 CA 2A A5:3C
A298:D6 01 CA 03 A5 D6 01 CA:24
A2A0:DC A4 D6 01 CA B5 A4 D6:92
A2A8:01 CA 8E A4 D6 01 CA 67:4F
A2B0:A4 D6 01 CA 40 A4 D6 01:52
A2B8:CA 19 A4 D6 01 CA F2 A3:17
A2C0:D6 01 CA CB A3 D6 01 CA:12
A2C8:A4 A3 D6 01 CA 7D A3 D6:48
A2D0:01 CA 56 A3 D6 01 CA 2F:06
A2D8:A3 D6 01 CA 08 A3 C3 E1:0D
A2E0:A2 3A B5 A1 CD 5C A2 32:B1
A2E8:B5 A1 3A B6 A1 CD 5C A2:3C
A2F0:32 B6 A1 3A B7 A1 CD 5C:D6
A2F8:A2 32 B7 A1 3A B8 A1 CD:26
A300:5C A2 32 B8 A1 C3 65 A2:F6
A308:3A B5 A1 CD 5C A2 32 B5:ED
A310:A1 3A B6 A1 CD 5C A2 32:E2
A318:B6 A1 3A B7 A1 CD 4A A2:5D
A320:32 B7 A1 3A B8 A1 CD 5C:09
A328:A2 32 B8 A1 C3 65 A2 3A:FC
A330:B5 A1 CD 53 A2 32 B5 A1:73
A338:3A B6 A1 CD 5C A2 32 B6:1F
A340:A1 3A B7 A1 CD 5C A2 32:13
A348:B7 A1 3A B8 A1 CD 4A A2:8F
A350:32 B8 A1 C3 65 A2 3A B5:37
A358:A1 CD 5C A2 32 B5 A1 3A:29
A360:B6 A1 CD 5C A2 32 B6 A1:AE
A368:3A B7 A1 CD 42 A2 32 B7:37
A370:A1 3A B8 A1 CD 5C A2 32:44
A378:B8 A1 C3 65 A2 3A B5 A1:CE
A380:CD 4A A2 32 B5 A1 3A B6:54
A388:A1 CD 53 A2 32 B6 A1 3A:51
A390:B7 A1 CD 5C A2 32 B7 A1:E0
A398:3A B8 A1 CD 4A A2 32 B8:71
A3A0:A1 C3 65 A2 3A B5 A1 CD:0B
A3A8:42 A2 32 B5 A1 3A B6 A1:48
A3B0:CD 5C A2 32 B6 A1 3A B7:98
A3B8:A1 CD 5C A2 32 B7 A1 3A:8B
A3C0:B8 A1 CD 4A A2 32 B8 A1:00
A3C8:C3 65 A2 3A B5 A1 CD 42:D4
A3D0:A2 32 B5 A1 3A B6 A1 CD:FB
A3D8:5C A2 32 B6 A1 3A B7 A1:94
A3E0:CD 5C A2 32 B7 A1 3A B8:CA
A3E8:A1 CD 42 A2 32 B8 A1 C3:2B
A3F0:65 A2 3A B5 A1 CD 53 A2:EC
A3F8:32 B5 A1 3A B6 A1 CD 53:D4
A400:A2 32 B6 A1 3A B7 A1 CD:2E
A408:53 A2 32 B7 A1 3A B8 A1:BE
A410:CD 53 A2 32 B8 A1 C3 65:29
A418:A2 3A B5 A1 CD 4A A2 32:D9
A420:B5 A1 3A B6 A1 CD 53 A2:6D
A428:32 B6 A1 3A B7 A1 CD 4A:FE
A430:A2 32 B7 A1 3A B8 A1 CD:60
A438:53 A2 32 B8 A1 C3 65 A2:26
A440:3A B5 A1 CD 42 A2 32 B5:0C
A448:A1 3A B6 A1 CD 4A A2 32:09
A450:B6 A1 3A B7 A1 CD 5C A2:A8
A458:32 B7 A1 3A B8 A1 CD 42:28
A460:A2 32 B8 A1 C3 65 A2 3A:35
A468:B5 A1 CD 53 A2 32 B5 A1:AC
A470:3A B6 A1 CD 4A A2 32 B6:46
A478:A1 3A B7 A1 CD 53 A2 32:43
A480:B7 A1 3A B8 A1 CD 42 A2:C0
A488:32 B8 A1 C3 65 A2 3A B5:70
A490:A1 CD 53 A2 32 B5 A1 3A:59
A498:B6 A1 CD 42 A2 32 B6 A1:CD
A4A0:3A B7 A1 CD 53 A2 32 B7:81
A4A8:A1 3A B8 A1 CD 53 A2 32:74
A4B0:B8 A1 C3 65 A2 3A B5 A1:07
A4B8:CD 42 A2 32 B5 A1 3A B6:85
```

```
A4C0:A1 CD 4A A2 32 B6 A1 3A:81
A4C8:B7 A1 CD 53 A2 32 B7 A1:10
A4D0:3A B8 A1 CD 42 A2 32 B8:A2
A4D8:A1 C3 65 A2 3A B5 A1 CD:44
A4E0:42 A2 32 B5 A1 3A B6 A1:81
A4E8:CD 4A A2 32 B6 A1 3A B7:BF
A4F0:A1 CD 42 A2 32 B7 A1 3A:AA
A4F8:B8 A1 CD 53 A2 32 B8 A1:42
A500:C3 65 A2 3A B5 A1 CD 42:0E
A508:A2 32 B5 A1 3A B6 A1 CD:35
A510:4A A2 32 B6 A1 3A B7 A1:BC
A518:CD 42 A2 32 B7 A1 3A B8:EA
A520:A1 CD 42 A2 32 B8 A1 C3:65
A528:65 A2 3A B5 A1 CD 42 A2:15
A530:32 B5 A1 3A B6 A1 CD 42:FD
A538:A2 32 B6 A1 3A B7 A1 CD:67
A540:42 A2 32 B7 A1 3A B8 A1:E6
A548:CD 42 A2 32 B8 A1 C3 65:51
A550:A2 3A B5 A1 4F CD A4 A1:88
A558:79 CD 9B A1 C9 3A B6 A1:D9
A560:18 F2 3A B7 A1 18 ED 3A:E0
A568:B8 A1 18 E8 00 00 00 00:66
```

```
10 MEMORY &9FFF:LOAD "harda06c.bin"  [2132]
:BORDER 0:INK 0,0
20 DIM PAL(14):FOR I=0 TO 14:READ P  [3114]
AL(I):INK I+1,PAL(I):NEXT
30 DATA 3,6,15,24,25,22,21,12,13,1,  [1260]
2,5,4,14,26
40 MODE 0:CLG 0:PAPER 0:PEN 14       [1111]
50 ORIGIN 320,250                    [237]
60 FOR J=48 TO 108:I=J*PI/52:MOVE C  [4133]
OS(I)*220,SIN(I)*160-40:DRAW 0,0,J
MOD 13+1:NEXT
70 FOR J=3 TO 23:FOR K=-110 TO 110   [6582]
STEP 220:I=J*PI/26:MOVE COS(I)*100+
K,SIN(I)*80-20:DRAW 0,0,(J+((K=110)
AND 4)-2) MOD 13+1:NEXT K,J
80 FOR J=22 TO 26:I=J*PI/52:MOVE CO  [5923]
S(I)*50,SIN(I)*120:DRAW 0,0,J MOD 1
3+1:NEXT
90 PRINT CHR$(22);CHR$(1);:LOCATE 4  [4359]
,2:PRINT"*** BONJOUR ***":LOCATE 7,
4:PRINT"LES  AMIS";CHR$(22);CHR$(0)
100 ORIGIN 112,8,112,528,36,8:CLG 1  [1682]
4
110 MOVE 0,0:DRAW 0,28,15:DRAW 420,  [2920]
28:DRAW 420,0:DRAW 0,0
120 MOVE 20,20:TAG#1:PRINT#1,"* SAL  [2933]
UT  ! *";
130 CALL &A06C                       [778]
```

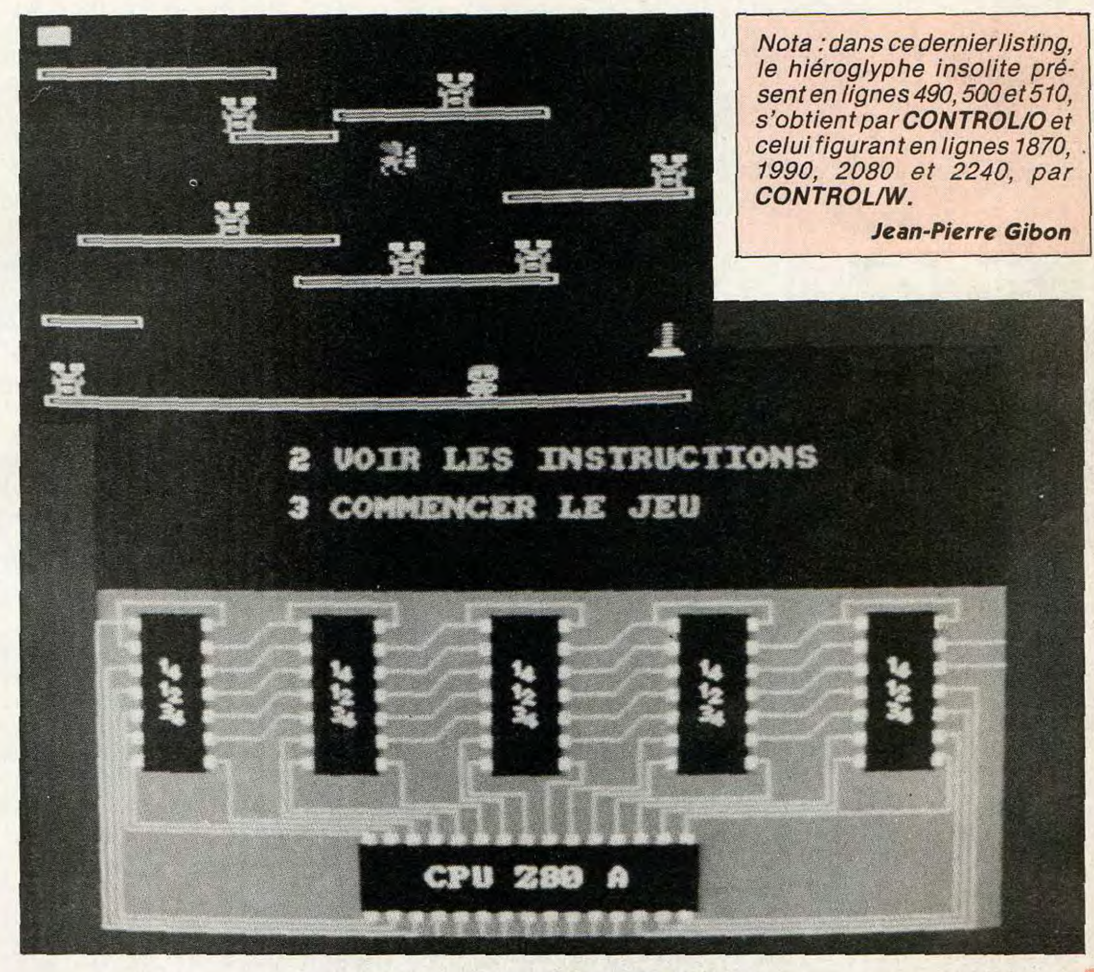
```
9500 'additif a "HARDA06C.BIN" pour  [4812]
 parametrage de fenetre
9510 ' CANOUIL Ph   octobre 1987     [1383]
9520 '                               [117]
9530 MEMORY &9FFF:ORIGIN 0,0,0,639,  [5499]
0,399:ax=PEEK(&B7E7):POKE &B7E7,11:
IF PEEK(&A000)<>&96 OR PEEK(&A001)<
>&A2 THEN LOAD "harda06c.bin"
9540 IF yma=0 OR xma=0 THEN ymi=177  [3738]
:yma=225:xmi=276:xma=328
9550 MOVE xmi-1,ymi-6,1,1::DRAW xma  [6928]
,ymi-6:DRAW xma,yma+2:DRAW xmi-1,ym
a+2:DRAW xmi-1,ymi-6:xmi1=xmi-1:ymi
1=ymi-6:xma1=xma:yma1=yma+2
9560 IF INKEY(2)=0 OR JOY(0)=2 THEN  [1762]
 IF ymi-6>=0 THEN ymi=ymi-6:GOSUB 9
660
9570 IF INKEY(0)=0 OR JOY(0)=1 THEN  [4890]
 IF yma+6<=399 THEN yma=yma+6:GOSUB
 9660
9580 IF INKEY(1)=0 OR JOY(0)=8 THEN  [3746]
 IF xma+4<=636 THEN xma=xma+4:GOSUB
 9660
9590 IF INKEY(8)=0 OR JOY(0)=4 THEN  [1478]
 IF xmi-4>=0 THEN xmi=xmi-4:GOSUB 9
660
9600 IF INKEY(2)=32 OR JOY(0)=18 TH  [2899]
EN IF ymi+6<=yma THEN ymi=ymi+6:GOS
UB 9660
9610 IF INKEY(0)=32 OR JOY(0)=17 TH  [2820]
EN IF yma-6>=ymi THEN yma=yma-6:GOS
UB 9660
9620 IF INKEY(1)=32 OR JOY(0)=24 TH  [2942]
EN IF xma-4>=xmi THEN xma=xma-4:GOS
UB 9660
9630 IF INKEY(8)=32 OR JOY(0)=20 TH  [2346]
EN IF xmi+4<=xma THEN xmi=xmi+4:GOS
UB 9660
9640 IF INKEY(6)=0 OR INKEY(18)=0 T  [1448]
HEN 9670
9645 IF INKEY(9)=0 THEN cop=1:GOTO   [1986]
9670 ELSE cop=0
9650 GOTO 9560                       [320]
9660 MOVE xmi1,ymi1,1,1:DRAW xma1,y  [12700]
mi1:DRAW xma1,yma1:DRAW xmi1,yma1:D
RAW xmi1,ymi1:FRAME:MOVE xmi-1,ymi-
6:DRAW xma,ymi-6:DRAW xma,yma+2:DRA
W xmi-1,yma+2:DRAW xmi-1,ymi-6:xmi1
=xmi-1:ymi1=ymi-6:xma1=xma:yma1=yma
+2:RETURN
9670 '                               [117]
9680 xmih=INT(xmi/256):xmib=xmi-(xm  [10840]
ih*256):xmah=INT(xma/256):xmab=xma-
(xmah*256):ymih=INT(ymi/256):ymib=y
mi-(ymih*256):ymah=INT(yma/256):yma
b=yma-(ymah*256)
9690 POKE &A141,&C3:POKE &A142,&6C:  [6687]
POKE &A143,&A5:POKE &A144,0:POKE &A
145,0:FOR i=&A56C TO &A58B:READ z:P
OKE i,z:NEXT
9700 POKE &A147,xmib:POKE &A148,xmi  [9447]
h:POKE &A0FB,xmib:POKE &A0FC,xmih:P
OKE &A134,xmab:POKE &A135,xmah:POKE
 &A0F5,ymab:POKE &A0F6,ymah:POKE &A
580,ymih:POKE &A585,ymib
9710 xxl=xma-xmi:xxlh=INT(xxl/256):  [7483]
xxlb=xxl-(xxlh*256):IF xxlb>=128 AN
D xxlb<256 THEN xxlh=xxlh+1:xxr=257
-xxlb:xxlb=0 ELSE xxr=0
9720 POKE &A56E,xxr:POKE &A574,xxr:  [8385]
POKE &A191,xxlb:POKE &A196,xxlh:POK
E &B7E7,ax:CLEAR INPUT:LOCATE 1,24:
IF cop=0 THEN END ELSE CALL &A06C:E
ND
9730 DATA &c5,&3e,0,&e6,&ff,&28,&09  [5309]
,&06,&00,&3e,0,&cd,&9b,&a1,&10,&f9,
&c1,&23,&7c,&fe,0,&20,&04,&7d,&fe,0
,&d8,&2b,&c3,&45,&a1,0
```



LA BECANE EN FOLIE

Jadis laborieux et fringant, votre ordinateur semble en proie à un mal mystérieux mettant en doute le bien fondé de sa réputation. *Appelé de toute urgence auprès de l'illustre patient, un membre microscopique du CROTTE (Centre de Réparation d'Ordinateur Teigneux et à Touches Explosives) ne tarde pas à découvrir la source du mal : les circuits pollués de microbes et les errances morbides de belliqueux protons à têtes chercheuses, perturbent à l'évidence le précieux mécanisme qu'il convient d'assainir sans tarder.*

*Sauvez le premier listing sous un nom de votre choix (« **BECANE** » semble approprié). Lancez les deuxième et troisième listings afin de créer les routines binaires **BECANE1.BIN** et **BECANE2.BIN** (vérification et sauvegarde automatiques) nécessaires au bon fonctionnement du programme principal. Sauvez finalement le quatrième listing (règles + programme principal) sous le nom « **BECANE3** »*

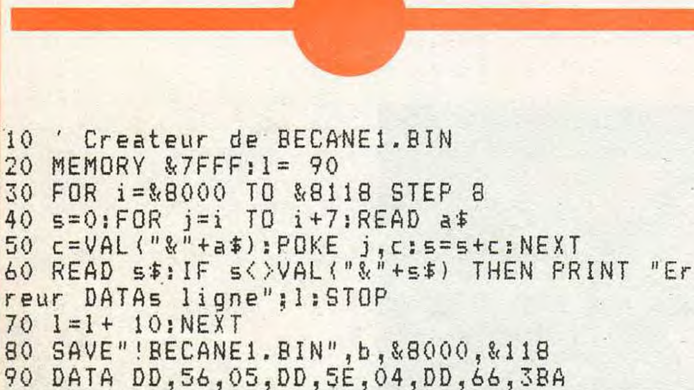


*Nota : dans ce dernier listing, le hiéroglyphe insolite présent en lignes 490, 500 et 510, s'obtient par **CONTROL/O** et celui figurant en lignes 1870, 1990, 2080 et 2240, par **CONTROL/W**.*

Jean-Pierre Gibon

```
10 '###################          [669]
20 '##  LA BECANE      ##        [1491]
30 '##       EN FOLIE  ##        [693]
40 '##---------------##          [1419]
50 '## GIBON J-Pierre ##         [1614]
60 '##    01.10.1986    ##       [746]
70 '###################          [669]
80 DEFINT a-z                    [553]
90 MODE 1:BORDER 0:INK 0,0:INK 1,26 [2736]
:INK 2,14:INK 3,7
100 PEN 2:LOCATE 12,2:PRINT"GIBON J [1722]
ean-Pierre"
110 PEN 1:LOCATE 13,5:PRINT"P r e s [2289]
 e n t e"
120 OUT &BC00,6:OUT &BD00,24     [902]
130 PRINT CHR$(23)CHR$(3):LOCATE 1, [3430]
25:PRINT"LA BECANE"
140 x=30:y=180:GOSUB 270         [711]
150 LOCATE 1,25:PRINT"EN FOLIE "  [2403]
160 x=230:y=80:GOSUB 270         [1133]
170 LOCATE 1,1:PRINT CHR$(23)CHR$(0 [968]
)
180 LOCATE 1,25:PRINT CHR$(20)   [1683]
190 OUT &BC00,6:OUT &BD00,25     [901]
200 INK 3,9,7                    [78]
210 PEN 2:LOCATE 10,25:PRINT"le pro [3351]
gramme se charge"
220 MEMORY &7FFF                 [150]
230 LOAD"!becane1.bin",&8000     [1546]
240 LOAD"!becane2.bin",&9000     [1170]
250 RUN"!becane3.bas"            [1218]
260 '--------------              [878]
270 FOR i=0 TO 140 STEP 2        [1354]
280 FOR j=0 TO 14 STEP 2         [765]
```

```
290 IF TEST(i,j) THEN 310                [1044]
300 NEXT:NEXT:RETURN                     [1435]
310 PLOT x+i*3,y+j*6,1                   [969]
320 DRAWR 0,14:DRAWR 8,0:DRAWR 0,-1 [2012]
4:DRAWR -8,0
330 PLOTR 2,2,3                          [480]
340 DRAWR 0,10:DRAWR 4,0:DRAWR 0,-1 [2174]
0:DRAWR -2,0:DRAWR 0,8
350 GOTO 300                             [504]
```



```
10 ' Createur de BECANE1.BIN
20 MEMORY &7FFF:l= 90
30 FOR i=&8000 TO &8118 STEP 8
40 s=0:FOR j=i TO i+7:READ a$
50 c=VAL("&"+a$):POKE j,c:s=s+c:NEXT
60 READ s$:IF s<>VAL("&"+s$) THEN PRINT "Er
reur DATAs ligne";l:STOP
70 l=l+ 10:NEXT
80 SAVE"!BECANE1.BIN",b,&8000,&118
90 DATA DD,56,05,DD,5E,04,DD,66,3BA
100 DATA 03,DD,6E,02,CD,1D,BC,DD,3D3
110 DATA 56,01,DD,5E,00,EB,F3,06,376
120 DATA 10,C5,D5,06,04,4E,1A,A9,2C5
130 DATA 12,23,13,10,F8,D1,E5,EB,3F1
140 DATA CD,26,BC,EB,E1,C1,10,E9,535
150 DATA FB,C9,DD,56,03,DD,5E,02,437
160 DATA DD,66,01,DD,6E,00,CD,1D,379
170 DATA BC,11,A0,00,19,AF,46,B0,32B
180 DATA 23,46,B0,23,46,B0,23,46,29B
190 DATA B0,32,00,A0,C9,AF,32,10,33C
200 DATA A0,DD,46,04,DD,7E,06,05,32D
210 DATA 05,05,05,05,05,05,05,90,0B3
220 DATA C8,D8,DD,46,04,DD,7E,06,428
230 DATA 04,04,04,04,04,04,90,C8,170
240 DATA D0,DD,46,00,DD,7E,02,05,355
250 DATA 05,05,05,05,05,05,05,05,028
260 DATA 05,05,05,90,C8,D8,DD,46,362
270 DATA 00,DD,7E,02,04,04,04,04,16D
280 DATA 04,04,04,04,04,04,04,04,020
290 DATA 90,C8,D0,3E,FF,32,10,A0,447
300 DATA C9,3E,00,32,11,A0,DD,66,32D
310 DATA 01,DD,6E,00,DD,56,03,DD,35F
320 DATA 5E,02,CD,1D,BC,7E,32,11,2C7
330 DATA A0,C9,01,50,00,C5,11,F0,380
340 DATA C0,21,F1,C0,ED,B0,C1,C5,5B5
350 DATA 11,F0,C8,21,F1,C8,ED,B0,540
360 DATA C1,C5,11,F0,D0,21,F1,D0,539
370 DATA ED,B0,C1,C5,11,F0,D8,21,51D
380 DATA F1,D8,ED,B0,C1,C5,11,F0,5ED
390 DATA E0,21,F1,E0,ED,B0,C1,C5,5F5
400 DATA 11,F0,E8,21,F1,E8,ED,B0,580
410 DATA C1,C5,11,F0,F0,21,F1,F0,579
420 DATA ED,B0,C1,11,F0,F8,21,F1,569
430 DATA F8,ED,B0,C9,00,00,00,00,35E
440 DATA 00,00,00,00,00,00,00,00,000
```

```
10 ' Createur de BECANE2.BIN
20 MEMORY &8FFF:l= 90
30 FOR i=&9000 TO &9350 STEP 8
40 s=0:FOR j=i TO i+7:READ a$
50 c=VAL("&"+a$):POKE j,c:s=s+c:NEXT
60 READ s$:IF s<>VAL("&"+s$) THEN PRINT "Er
reur DATAs ligne";l:STOP
70 l=l+ 10:NEXT
80 SAVE"!BECANE2.BIN",b,&9000,&350
90 DATA 10,50,00,00,30,70,A0,00,1A0
100 DATA 30,30,A0,00,30,30,A0,40,240
110 DATA 10,30,A0,40,00,30,A0,80,270
120 DATA 00,10,E4,CC,01,12,44,8C,2A3
130 DATA 03,12,46,CC,03,03,46,CF,242
140 DATA 01,03,44,00,51,51,44,CF,1FD
150 DATA A2,51,00,CC,A2,51,00,00,2B2
160 DATA A2,00,A2,00,51,00,51,00,1E6
170 DATA 00,00,00,00,00,00,00,00,000
180 DATA 00,00,00,00,00,00,00,40,040
190 DATA 00,00,00,40,00,00,00,80,0C0
200 DATA 00,00,44,CC,01,03,44,8C,1E4
210 DATA 03,12,46,CC,03,30,46,CF,26F
220 DATA 10,30,E4,00,30,30,E4,CF,337
230 DATA 30,30,A0,CC,B2,71,00,00,2EF
240 DATA A2,00,A2,00,51,00,51,00,1E6
250 DATA 00,00,A0,20,00,50,B0,30,1F0
260 DATA 00,50,30,30,80,50,30,30,1E0
270 DATA 80,50,30,20,40,50,30,00,1E0
280 DATA CC,D8,20,00,4C,88,21,02,2B8
290 DATA CC,89,21,03,CF,89,03,03,2D7
300 DATA 00,88,03,02,CF,88,A2,A2,328
310 DATA CC,00,A2,51,00,00,A2,51,2B2
320 DATA 00,51,00,51,00,A2,00,A2,1E6
330 DATA 00,00,00,00,00,00,00,00,000
340 DATA 00,00,00,00,80,00,00,00,080
350 DATA 80,00,00,00,40,00,00,00,0C0
360 DATA CC,88,00,00,4C,88,03,02,22D
370 DATA CC,89,21,03,CF,89,30,03,304
380 DATA 00,D8,30,20,CF,D8,30,30,32F
390 DATA CC,50,30,30,00,00,B2,71,29F
400 DATA 00,51,00,51,00,A2,00,A2,1E6
410 DATA 00,FC,A8,00,54,F3,F6,FC,4DD
420 DATA F9,F6,82,00,F9,E9,20,00,473
430 DATA FC,C3,C3,00,41,C3,82,00,408
440 DATA 41,82,00,00,00,C3,82,00,208
450 DATA 00,04,00,00,00,F0,A0,C0,254
460 DATA 00,E1,C3,82,00,F0,A0,00,3B6
470 DATA 00,F0,A0,00,00,50,00,00,1E0
480 DATA 00,50,00,00,00,50,A0,00,140
490 DATA 00,FC,A8,00,54,F3,FC,FC,4E3
500 DATA F9,F6,82,00,F9,E9,20,00,473
510 DATA FC,C3,C3,00,41,C3,82,00,408
520 DATA 41,82,00,00,00,C3,82,00,208
530 DATA 00,04,00,00,00,F0,A0,C0,254
540 DATA 00,E1,C3,82,00,F0,A0,00,3B6
550 DATA 00,F0,A0,00,00,F0,A0,A0,3C0
560 DATA 50,A0,F0,A0,50,00,50,00,320
570 DATA 00,54,FC,00,FC,F9,F3,A8,4E0
580 DATA 00,41,F9,F6,00,10,D6,F6,40C
590 DATA 00,C3,C3,FC,00,41,C3,82,408
600 DATA 00,00,41,82,00,41,C3,00,1C7
610 DATA 00,00,08,00,C0,50,F0,00,208
620 DATA 41,C3,D2,00,00,50,F0,00,316
630 DATA 00,50,F0,00,00,00,A0,00,1E0
640 DATA 00,00,A0,00,00,50,A0,00,190
650 DATA 00,54,FC,00,FC,F9,F3,A8,4E0
660 DATA 00,41,F9,F6,00,10,D6,F6,40C
670 DATA 00,C3,C3,FC,00,41,C3,82,408
680 DATA 00,00,41,82,00,41,C3,00,1C7
690 DATA 00,00,08,00,C0,50,F0,00,208
```

```
700 DATA 41,C3,D2,00,00,50,F0,00,316
710 DATA 00,50,F0,00,50,50,F0,00,2D0
720 DATA 50,F0,50,A0,00,A0,00,A0,370
730 DATA 00,FC,FC,A8,54,F3,F3,F6,5D0
740 DATA 54,FC,FC,FC,41,86,8C,C3,55E
750 DATA 41,C3,C9,C3,41,41,C9,41,41C
760 DATA 41,82,00,C3,00,C3,C3,82,38E
770 DATA 00,00,08,00,00,D2,F0,82,24C
780 DATA 41,D2,F0,C3,41,50,F0,41,488
790 DATA 41,D2,F0,C3,00,D2,F0,82,50A
800 DATA 00,50,50,00,00,F0,50,A0,280
810 DATA 00,14,28,00,00,CC,CC,00,1D4
820 DATA 00,14,28,00,44,CC,CC,88,2A0
830 DATA 00,14,28,00,CC,CC,CC,CC,36C
840 DATA 00,14,28,00,44,CC,CC,88,2A0
850 DATA 00,14,28,00,00,CC,CC,00,1D4
860 DATA 00,14,28,00,00,30,30,00,09C
870 DATA 00,30,30,00,04,0C,0C,08,084
880 DATA 04,0C,0C,08,CF,CF,CF,CF,360
890 DATA CF,8A,45,CF,8A,8A,45,45,40B
900 DATA CF,8A,45,CF,14,00,00,28,2A9
910 DATA 14,28,14,28,00,3C,3C,00,0F0
920 DATA 22,14,28,11,22,6C,9C,11,1AA
930 DATA 3C,CD,CE,3C,3C,00,00,3C,28B
940 DATA 14,00,00,28,14,CD,CE,28,213
950 DATA 14,6C,9C,28,14,14,28,28,1BC
960 DATA 14,00,00,28,33,00,00,33,0A2
970 DATA 00,00,44,00,00,00,01,00,045
980 DATA 44,00,01,00,01,00,02,44,08C
990 DATA 01,00,02,01,00,02,02,02,00A
1000 DATA 88,01,C0,00,03,47,CF,02,264
1010 DATA 00,45,CF,44,00,42,81,00,21B
1020 DATA 01,00,02,02,88,01,88,01,117
1030 DATA 00,01,00,44,00,88,00,00,0CD
1040 DATA 00,00,00,00,00,00,00,00,000
1050 DATA 20,10,88,00,88,44,00,10,194
1060 DATA 98,44,00,CC,44,00,01,00,1ED
1070 DATA 01,02,02,10,00,02,02,44,05D
1080 DATA 20,02,88,03,88,00,44,02,17B
1090 DATA 01,00,00,00,00,02,01,00,004
1100 DATA 01,02,02,02,89,00,02,CC,15E
1110 DATA 98,00,88,10,00,44,20,00,194
1120 DATA 00,00,00,00,00,00,00,00,000
1130 DATA 00,00,00,00,00,00,00,00,000
1140 DATA 00,00,00,00,00,00,00,00,000
1150 DATA 00,00,00,00,00,00,00,00,000
```

```
10 '###################                    [669]
20 '##  La becane     ##                   [1491]
30 '##      en folie  ##                   [693]
40 '##---------------##                    [1419]
50 '## GIBON J-Pierre ##                   [1614]
60 '##   01.10.1986   ##                   [746]
70 '###################                    [669]
80 '                                       [117]
90 ' ********************                  [665]
100 '*  initialisation  *                  [936]
110 '********************                  [665]
120 DEFINT a-z                             [553]
130 DATA &9000,&9040,&9080,&90C0,&9 [3290]
100,&9140,&9180,&91C0,&9200,&9240,&
9280,&92C0,&9300
140 RESTORE 130:DIM spr(12)         [1532]
150 FOR i=0 TO 12:READ spr(i):NEXT  [2080]
160 ENV 1,15,-1,1:ENV 2,1,-2,1,13,- [6113]
1,4:ENV 13,15,1,1,1,0,5,15,-1,2:ENV
 14,1,5,1,10,1,4,5,-3,1:ENV 15,1,5,
1,10,1,4,5,-3,1
170 ENT 1,10,-30,1,10,10,1:ENT 2,60 [7887]
,-20,1:ENT 3,60,20,1:ENT 4,40,-3,1:
ENT 5,200,1,1:ENT 13,20,-1,1,1,0,10
,60,1,1:ENT 14,40,-10,1:ENT 15,5,-2
0,1,60,10,1
180 SYMBOL 255,6,8,60,102,126,96,60 [1758]
190 '********************           [665]
200 '*  dessin circuits *           [1352]
210 '********************           [665]
220 SYMBOL 252,60,60,60:SYMBOL 253, [5886]
0,0,0,0,0,60,60,60:SYMBOL 254,0,0,2
24,224,224,224:SYMBOL 255,0,0,7,7,7
,7
230 INK 0,0:INK 1,24:INK 2,26:INK 3 [1976]
,9:BORDER 1:MODE 1
240 PEN 0:PAPER 3:LOCATE 1,11:PRINT [1896]
 CHR$(20)
250 FOR i=3 TO 35 STEP 8:WINDOW#1,i [3049]
,i+2,12,18:CLS#1:NEXT
260 WINDOW#1,13,27,22,24:CLS#1      [1775]
270 PEN 2:FOR i=2 TO 34 STEP 8:FOR  [7022]
j=12 TO 18:LOCATE i,j:PRINT CHR$(25
5):LOCATE i+4,j:PRINT CHR$(254):NEX
T j,i
280 FOR i=13 TO 27:LOCATE i,21:PRIN [5858]
T CHR$(253):LOCATE i,25:PRINT CHR$(
252):NEXT
290 PAPER 0:LOCATE 16,23:PRINT"CPU  [1675]
Z80 A"
300 FOR i=4 TO 38 STEP 8:LOCATE i,1 [5730]
4:PRINT CHR$(168):LOCATE i,15:PRINT
 CHR$(169):LOCATE i,16:PRINT CHR$(1
70):NEXT
310 DATA -24,216,0,216,0,0,230,0,23 [7870]
0,8,-24,200,16,200,16,216,-24,184,6
,184,6,4,214,4,214,8,-198,8,12,8,12
,168,24,168,-24,152,18,152,18,70,19
8,70,-24,136,18,136,-24,120,24,74,2
14,74,214,70
320 DATA -230,70,230,78,96,78,96,12 [12184]
0,86,120,-246,70,246,82,136,82,136,
136,152,136,-152,120,142,120,142,86
,262,86,262,70,-278,70,278,90,224,9
0,224,120,214,120,-280,136,266,136,
266,94,294,94,294,70,-310,70,310,98
,272,98,272,120,280,120
330 DATA -326,70,326,106,350,106,35 [10776]
0,120,342,120,-342,70,342,102,390,1
02,390,136,408,136,-408,120,396,120
,396,98,358,98,358,70,-374,70,374,9
4,478,94,478,120,470,120,-390,70,39
0,90,518,90,518,136,536,136,-406,70
,406,86,524,86,524,120,536,120
340 DATA -422,70,422,82,602,82,602, [11118]
120,598,120,-598,136,608,136,608,8,
422,8,-406,8,406,4,614,4,614,152,59
8,152,-598,168,620,168,620,0,390,0,
390,8,-598,184,640,184,-598,200,640
,200,-16,216,16,230,94,230,94,216,8
6,216,1,1
350 DATA 64,16,34,48,18,32,32,20,30 [2568]
,16,22,28,0,24,26
360 RESTORE 310                     [749]
```

```
370 READ a,b:IF a<0 THEN MOVE -a,b: [1613]
GOTO 370
380 IF a<>1 THEN DRAW a,b,1:GOTO 37 [2193]
0
390 FOR i=246 TO 374 STEP 16:PLOT i [2638]
,0:DRAW i,8:NEXT
400 FOR i=144 TO 528 STEP 128:ORIGI [1794]
N i,216
410 PLOT 0,0:DRAWR 0,14:DRAWR 78,0: [1450]
DRAWR 0,-14:DRAWR -8,0
420 RESTORE 350:FOR j=1 TO 5        [1092]
430 READ a,b,c:PLOT 8,-a:DRAWR -b,0 [3442]
:DRAWR -16,-16:DRAWR -c,0:NEXT j,i
440 ORIGIN 0,0                      [298]
450 '*******************            [665]
460 '*       menu      *            [551]
470 '*******************            [665]
480 WINDOW 1,40,1,10:CLS            [1595]
490 LOCATE 10,3:PRINT"Θ2""1 Θ1VOIR  [2737]
LE PARCOURS"
500 LOCATE 10,5:PRINT"Θ2""2 Θ1VOIR  [3262]
LES INSTRUCTIONS"
510 LOCATE 10,7:PRINT"Θ2""3 Θ1COMME [3255]
NCER LE JEU"
520 WHILE INKEY$<>"":WEND           [1786]
530 e$=INKEY$:IF e$="" THEN 530     [569]
540 IF e$="1" THEN CLS:GOTO 610     [1249]
550 IF e$="2" THEN 1150             [812]
560 IF e$="3" THEN 1650             [778]
570 GOTO 520                        [413]
580 '*******************            [665]
590 '*    parcours     *            [993]
600 '*******************            [665]
610 ci=1:x=390:y=8:dx=0:dy=-2:GOSUB [3227]
 750:wi=35:GOSUB 970
620 x=536:y=136:dx=-2:dy=0:GOSUB 75 [2902]
0
630 ci=2:x=374:y=70:dx=0:dy=2:GOSUB [3164]
 750:wi=27:GOSUB 970
640 x=408:y=136:dx=-2:dy=0:GOSUB 75 [2354]
0
650 ci=3:x=326:y=70:dx=0:dy=2:GOSUB [3317]
 750:wi=19:GOSUB 970
660 x=280:y=136:dx=-2:dy=0:GOSUB 75 [1930]
0
670 ci=4:x=278:y=70:dx=0:dy=2:GOSUB [1341]
 750:wi=11:GOSUB 970
680 x=152:y=136:dx=-2:dy=0:GOSUB 75 [2258]
0
690 ci=5:x=230:y=70:dx=0:dy=2:GOSUB [3092]
 750:wi=3:GOSUB 970
700 x=24:y=184:dx=-2:dy=0:GOSUB 750 [2153]
710 GOTO 480                        [373]
720 '*******************            [665]
730 '*  deplacements   *            [986]
740 '*******************            [665]
750 FOR z=1 TO 400:NEXT:SOUND 1,200 [3931]
,4,7:IF dx=0 THEN 810
760 IF TEST(x+dx,y)=2 THEN 860 ' Bo [3521]
rne circuit integre
770 IF TEST(x+dx,y)<>1 THEN 870 ' C [3603]
hangement direction
780 CALL &BD19:PLOT x,y,1           [402]
790 x=x+dx:PLOT x,y,0               [1953]
800 GOTO 760                        [413]
810 IF TEST(x,y+dy)=2 THEN 860 ' Bo [3124]
rne circuit integre
820 IF TEST(x,y+dy)<>1 THEN 870 ' C [2407]
hangement direction
830 CALL &BD19:PLOT x,y,1           [402]
840 y=y+dy:PLOT x,y,0               [702]
850 GOTO 810                        [306]
860 PLOT x,y,1:SOUND 1,100,4,7:RETU [2091]
RN
870 IF dx=0 THEN 910                [725]
880 dx=0                            [61]
890 IF TEST(x,y-2)=1 THEN dy=-2:GOT [2312]
O 810
900 IF TEST(x,y+2)=1 THEN dy=2:GOTO [2527]
 810
910 dy=0                            [49]
920 IF TEST(x-2,y)=1 THEN dx=-2:GOT [1793]
O 760
930 IF TEST(x+2,y)=1 THEN dx=2:GOTO [1323]
 760
940 '*******************            [665]
950 '*     tableau     *            [378]
960 '*******************            [665]
970 PEN 2:LOCATE 17,6:PRINT"Circuit [2547]
";ci
980 FOR z=1 TO 400:NEXT             [1530]
990 FOR i=1 TO 11                   [314]
1000 fond=fond XOR 2:WINDOW #1,wi,w [3666]
i+2,12,18:PAPER #1,fond:CLS#1
1010 SOUND 1,INT(RND*2000)+50,INT(R [2977]
ND*12)+1,0,1,,INT(RND*31)+1
1020 SOUND 1,0,8                    [1069]
1030 NEXT                           [350]
1040 WINDOW #1,wi,wi+2,12,18:PAPER  [2208]
#1,0:CLS#1
1050 PEN #1,2                       [472]
1060 LOCATE #1,2,3:PRINT #1,CHR$(16 [1929]
8)
1070 LOCATE #1,2,4:PRINT #1,CHR$(16 [2180]
9)
1080 LOCATE #1,2,5:PRINT #1,CHR$(17 [2593]
0)
1090 FOR z=1 TO 600:NEXT            [1230]
1100 LOCATE 1,6:PRINT CHR$(18)      [1465]
1110 RETURN                         [555]
1120 '*******************           [665]
1130 '*   instructions  *           [214]
1140 '*******************           [665]
1150 MODE 2                         [513]
1160 INK 1,0:INK 0,13:BORDER 13     [1384]
1170 PRINT"INSTRUCTIONS :":PRINT    [2227]
1180 DATA "Vous etes membre du CROT [18671]
TE","( Centre de Reparation d'Ordin
ateurs Teigneux et a Touches Explos
ives )", ,"Votre ordinateur est ent
ierement deglingue ( D'ailleurs, il
 n'est pas le seul )"
1190 DATA "Apres tout, dans ce jeu  [11961]
(genial!) , vous ne serez qu'un neu
tron ! MINUS !!","Et vous devrez la
 refoutre sur pied, cette sal... de
becane, en zigouillant"
1200 DATA "tous les affreux microbe [6389]
s qui occupent les circuits integre
s."
1210 DATA  ,"Enfin bref, tout ca po [15632]
ur vous dire","de faire quand-meme
attention aux protons baladeurs a t
ete chercheuse","et a bien d'autres
 fantaisies de l'auteur !"
1220 DATA  ,"Un teleporteur vous pe [5215]
rmettra de passer d'un circuit a un
 autre."
1230 DATA  ,"Mais maintenant, passe [15173]
z a l'acte. Non non, pas celui-la v
```

```
icieux !","Je voulais dire: au boul
ot !", ,"Fais gaffe quand-meme ! He
in ! Ca s'ra pas d'la tarte, mon ga
rs !"
1240 DATA "*"                          [183]
1250 RESTORE 1180                      [747]
1260 READ a$:IF a$="*" THEN 1330       [1003]
1270 FOR i=1 TO LEN(a$)                [791]
1280 PRINT MID$(a$,i,1);               [1271]
1290 SOUND 1,INT(RND*400)+20,1,5       [1658]
1300 FOR z=1 TO 70:NEXT                [1136]
1310 NEXT:PRINT                        [811]
1320 GOTO 1260                         [351]
1330 LOCATE 18,25:PRINT"Quand vous     [5323]
voulez pour appuyer sur une touche
!"
1340 WHILE INKEY$<>"":WEND:CALL &BB [2497]
06
1341 MODE 1:BORDER 0:INK 0,0:INK 1, [2108]
14:INK 2,26:INK 3,6
1342 PRINT"Les touches utilisees so [3868]
nt:"
1343 PEN 3:LOCATE 15,5:PRINT CHR$(2 [2067]
42)
1344 LOCATE 19,5:PRINT"COPY"           [2009]
1345 LOCATE 26,5:PRINT CHR$(243)       [1331]
1346 PEN 2:LOCATE 10,7:PRINT"gauche [2916]
 sauter  droite"
1347 PEN 1:LOCATE 1,10:PRINT"Deroul [2661]
ement du jeu:"
1348 PEN 3:PRINT                       [747]
1349 PRINT"  Une fois les microbes    [11838]
abattus,","  un teleporteur apparai
tra sur l'ecran,  vous permettant a
insi de passer au","  circuit suiva
nt."
1350 PRINT:PRINT"  Faites attention [11410]
 au proton baladeur      car sa vite
sse accelere en fonction","  des ta
bleaux (circuits)."
1358 WHILE INKEY$<>"":WEND:CALL &BB [2497]
06
1359 GOTO 1390                         [421]
1360 DATA 0,24,20,6,2,11,18,12,15,1 [1855]
6,17,26,8,13,4,3
1370 RESTORE 1360:FOR i=0 TO 15:REA [1429]
D a
1380 INK i,a:NEXT:RETURN               [1419]
1390 BORDER 0:MODE 0:GOSUB 1360        [1877]
1400 PEN 9:PRINT"PERSONNAGES & OBJE [2275]
TS"
1410 p=4:s=p:x=30:y=152:z=200:GOSUB [2942]
 1460
1420 PEN 1:LOCATE 7,8:PRINT"VOUS"      [1654]
1430 p=0:s=p:x=30:y=120:z=200:GOSUB [3093]
 1460
1440 LOCATE 7,12:PRINT"UN PROTON"      [1107]
1450 GOTO 1540                         [391]
1460 CALL &8000,0,y,spr(s)             [1015]
1470 FOR i=0 TO x STEP 2               [772]
1480 p=p XOR 1:IF p=4 THEN SOUND 1, [1723]
0,1,4,,,10
1490 CALL &8000,i,y,spr(s)             [516]
1500 CALL &8000,i+2,y,spr(p)           [1055]
1510 FOR t=1 TO z:NEXT                 [1372]
1520 s=p:NEXT                          [530]
1530 RETURN                            [555]
1540 FOR z=1 TO 1500:NEXT              [1494]
1550 CALL &8000,32,88,&9280            [1150]
1560 LOCATE 7,16:PRINT"UN MICROBE"     [1848]
1570 CALL &8000,32,56,&9240            [967]
1580 LOCATE 7,20:PRINT"TELEPORTEUR"    [1539]
1590 GOTO 1600                         [347]
1600 CALL &8000,32,152,spr(4)          [1267]
1610 CALL &8000,32,152,spr(8)          [1099]
1620 PEN 5:LOCATE 4,25:PRINT"<<< ES [3688]
PACE >>>"
1630 WHILE INKEY(47):WEND              [2005]
1640 GOTO 230                          [423]
1650 '*******************              [665]
1660 '*      debut jeu     *           [303]
1670 '*******************              [665]
1680 MODE 0:BORDER 0:GOSUB 1360        [1154]
1690 PEN 5:LOCATE 6,9:PRINT CHR$(24 [1667]
2);
1700 PEN 11:PAPER 5:PRINT" Z80"CHR$ [2905]
(149)"RAM ";
1710 PEN 5:PAPER 0:PRINT CHR$(243)     [2813]
1720 PEN 11:FOR i=10 TO 14             [1077]
1730 LOCATE 11,i:PRINT CHR$(149)       [1074]
1740 NEXT                              [350]
1750 CALL &8000,150,128,spr(9)         [1658]
1760 PAPER 15:PRINT CHR$(20):PAPER     [2085]
0
1770 p=4:s=p:x=148:y=103:z=60:GOSUB [3293]
 1460
1780 CALL &8000,150,103,spr(5)         [1264]
1790 CALL &8000,150,103,spr(8)         [1286]
1800 x=600:y=224:z=48:GOSUB 1860       [1594]
1810 GOTO 2290                         [345]
1820 SOUND 1,300,20,4,,1:RETURN        [1464]
1830 '*******************              [665]
1840 '*    disparition     *           [1214]
1850 '*******************              [665]
1860 SOUND 1,1500,60,5,,2              [1084]
1870 PRINT"⊣1":FOR i=0 TO z STEP 2     [1506]
1880 ON SQ(1) GOSUB 1820               [988]
1890 MOVE x,y-i:DRAWR 30,0,5           [1393]
1900 FOR t=1 TO 60:NEXT:NEXT           [1834]
1910 FOR i=z TO 0 STEP-2               [742]
1920 ON SQ(1) GOSUB 1820               [988]
1930 MOVE x,y-i:DRAWR 30,0,9           [1373]
1940 FOR t=1 TO 60:NEXT:NEXT           [1834]
1950 FOR i=0 TO z STEP 2               [708]
1960 ON SQ(1) GOSUB 1820               [988]
1970 MOVE x,y-i:DRAWR 30,0,7           [1391]
1980 FOR t=1 TO 60:NEXT:NEXT           [1834]
1990 PRINT"⊣0":FOR i=z TO 0 STEP-2     [1555]
2000 ON SQ(1) GOSUB 1820               [988]
2010 MOVE x,y-i:DRAWR 30,0,0           [1382]
2020 FOR t=1 TO 60:NEXT:NEXT           [1834]
2030 SOUND 129,60,70,5,,3:RETURN       [1929]
2040 '*******************              [665]
2050 '*    apparition      *           [699]
2060 '*******************              [665]
2070 SOUND 1,1500,60,5,,2              [1084]
2080 PRINT"⊣1":FOR i=0 TO z STEP 2     [1506]
2090 ON SQ(1) GOSUB 1820               [988]
2100 MOVE x,y-i:DRAWR 30,0,5           [1393]
2110 FOR t=1 TO 60:NEXT:NEXT           [1834]
2120 FOR i=z TO 0 STEP-2               [742]
2130 ON SQ(1) GOSUB 1820               [988]
2140 MOVE x,y-i:DRAWR 30,0,9           [1373]
2150 FOR t=1 TO 60:NEXT:NEXT           [1834]
2160 FOR i=0 TO z STEP 2               [708]
2170 MOVE x,y-i:DRAWR 30,0,7           [1391]
2180 FOR t=1 TO 60:NEXT:NEXT           [1834]
2190 CALL &8000,ax,ay,spr(8)           [1792]
2200 FOR i=z TO 0 STEP-2               [742]
2210 ON SQ(1) GOSUB 1820               [988]
```

```
2220 MOVE x,y-i:DRAWR 30,0,11          [1497]
2230 FOR t=1 TO 60:NEXT:NEXT           [1834]
2240 PRINT"┤0"                         [351]
2250 SOUND 129,80,60,5,,3:RETURN       [1918]
2260 '*******************              [665]
2270 '*      tableau 1     *           [421]
2280 '*******************              [665]
2290 tabl=1:GOSUB 3770                 [1522]
2300 DATA 0,0,638,0,80,100,40,160,3 [4063]
00,260,120,510,460,200,638,300,280,
500,200,260,300,0,320,240,*
2310 DATA 46,100,88,80,118,80,150,1 [2363]
20,100,160,48,150,2,20,*
2320 RESTORE 2300:col=10:co2=13:GOS [2179]
UB 3040
2330 GOSUB 3120                        [969]
2340 kx=150:ky=38                      [1171]
2350 kx1=2:ky1=199                     [919]
2360 CALL &8000,kx,ky,spr(9)           [1671]
2370 ax=150:ay=20:x=600:y=44:z=34:G [1678]
OSUB 2070
2380 sx=2:sy=180:sp=8:face=1:sc=7:G [2237]
OSUB 3210
2390 ax=150:ay=40:x=8:y=366:z=36:GO [2786]
SUB 1860
2400 z=REMAIN(0):WHILE INKEY$<>"":W [2561]
END
2410 '*******************              [665]
2420 '*      tableau 2     *           [411]
2430 '*******************              [665]
2440 tabl=2:GOSUB 3770                 [1822]
2450 DATA 0,0,638,220,80,420,160,16 [4083]
0,260,380,160,480,100,240,200,260,2
40,520,40,320,300,480,320,638,*
2460 DATA 144,20,114,100,124,140,64 [3423]
,140,40,100,24,140,56,180,10,180,*
2470 RESTORE 2450:col=2:co2=5:GOSUB [879]
 3040
2480 GOSUB 3120                        [969]
2490 kx=2:ky=38                        [768]
2500 kx1=150:ky1=199                   [622]
2510 CALL &8000,kx,ky,spr(9)           [1671]
2520 ax=2:ay=20:x=8:y=44:z=34:GOSUB [1799]
 2070
2530 sx=150:sy=180:sp=8:face=1:sc=8 [1540]
:GOSUB 3210
2540 ax=2:ay=40:x=600:y=366:z=36:GO [3020]
SUB 1860
2550 z=REMAIN(0):WHILE INKEY$<>"":W [2561]
END
2560 '*******************              [665]
2570 '*      tableau 3     *           [425]
2580 '*******************              [665]
2590 tabl=3:GOSUB 3770                 [1383]
2600 DATA 0,0,638,40,80,240,280,80, [6684]
380,440,80,590,100,160,300,400,160,
590,0,240,140,260,240,500,560,240,6
38,0,320,60,200,320,400,460,320,500
,*
2610 DATA 142,100,128,20,80,20,16,2 [2749]
0,2,140,84,140,50,180,100,100,*
2620 RESTORE 2600:col=3:co2=6:GOSUB [1865]
 3040
2630 GOSUB 3120                        [969]
2640 kx=150:ky=158                     [1212]
2650 kx1=2:ky1=199                     [919]
2660 CALL &8000,kx,ky,spr(9)           [1671]
2670 ax=150:ay=140:x=600:y=284:z=34 [2372]
:GOSUB 2070
2680 sx=2:sy=180:sp=8:face=1:sc=8:G [1669]
OSUB 3210
2690 ax=150:ay=80:x=8:y=366:z=36:GO [2292]
SUB 1860
2700 z=REMAIN(0):WHILE INKEY$<>"":W [2561]
END
2710 '*******************              [665]
2720 '*      tableau 4     *           [319]
2730 '*******************              [665]
2740 tabl=4:GOSUB 3770                 [1494]
2750 DATA 0,0,638,0,80,200,440,80,6 [4747]
38,160,160,290,350,160,480,450,240,
638,200,320,580,0,240,230,*
2760 DATA 28,20,2,60,66,100,2,140,6 [3113]
4,180,126,20,150,60,86,100,136,140,
96,180,*
2770 RESTORE 2750:col=1:co2=12:GOSU [1920]
B 3040
2780 GOSUB 3120                        [969]
2790 kx=80:ky=38                       [858]
2800 kx1=80:ky1=199                    [724]
2810 CALL &8000,kx,ky,spr(9)           [1671]
2820 ax=80:ay=20:x=320:y=44:z=34:GO [1074]
SUB 2070
2830 sx=80:sy=180:sp=8:face=1:sc=10 [2327]
:GOSUB 3210
2840 ax=80:ay=40:x=320:y=366:z=36:G [2083]
OSUB 1860
2850 z=REMAIN(0):WHILE INKEY$<>"":W [2561]
END
2860 '*******************              [665]
2870 '*      tableau 5     *           [317]
2880 '*******************              [665]
2890 tabl=5:GOSUB 3770                 [1282]
2900 DATA 0,0,638,0,320,140,190,320 [7030]
,550,600,320,638,140,240,300,390,24
0,500,550,240,638,0,160,200,400,160
,550,200,80,270,350,80,450,*
2910 DATA 62,140,36,100,2,100,18,20 [3774]
,54,20,96,20,128,20,118,100,150,140
,82,180,*
2920 RESTORE 2900:col=8:co2=4:GOSUB [3280]
 3040
2930 GOSUB 3120                        [969]
2940 kx=2:ky=199                       [949]
2950 kx1=150:ky1=38                    [652]
2960 CALL &8000,kx,ky,spr(9)           [1671]
2970 ax=2:ay=180:x=8:y=366:z=34:GOS [1860]
UB 2070
2980 sx=150:sy=20:sp=8:face=1:sc=10 [1485]
:GOSUB 3210
2990 ax=0:ay=198:x=600:y=44:z=34:GO [1416]
SUB 1860
3000 z=REMAIN(0):WHILE INKEY$<>"":W [3460]
END:GOTO 4740
3010 '*******************              [665]
3020 '*    aff. etages     *           [259]
3030 '*******************              [665]
3040 READ a$:IF a$="*" THEN RETURN     [1178]
3050 a=VAL(a$):READ b,c                [1416]
3060 PLOT a,b,col:DRAW c,b:DRAW c,b [3313]
+8:DRAW a,b+8:DRAW a,b
3070 PLOT a+8,b+4,co2:DRAW c-8,b+4     [1231]
3080 GOTO 3040                         [441]
3090 '*******************              [665]
3100 '*    aff. microbes   *           [1116]
3110 '*******************              [665]
3120 READ a$:IF a$="*" THEN RETURN     [1178]
3130 a=VAL(a$):READ b                  [1526]
3140 CALL &8000,a,b,spr(10)            [1260]
3150 GOTO 3120                         [457]
```

```
3160 '***********                        [332]
3170 '*         *                        [175]
3180 '*   JEU   *                        [230]
3190 '*         *                        [175]
3200 '***********                        [332]
3210 sc1=0:touch=0:bx=80:by=198:bbx [2287]
=bx:bby=by:pp=0:pp1=pp:ji=0
3220 POKE &A010,0                        [343]
3230 CALL &8000,bx,by,spr(0)             [849]
3240 EVERY 11-tabl GOSUB 3420            [1924]
3250 GOTO 3350                           [443]
3260 IF NOT INKEY(1) AND ax<150 THE [3830]
N ax=ax+2:k=4:tm=8:GOTO 3310
3270 IF NOT INKEY(8) AND ax>2 THEN  [3454]
ax=ax-2:k=6:tm=-2:GOTO 3310
3280 IF NOT INKEY(9) THEN 3530          [1139]
3290 IF face=1 THEN 3380                [1281]
3300 face=1:sp=8:GOTO 3320              [1266]
3310 face=0:ki=ki XOR 1:sp=k+ki:IF  [3661]
ki THEN SOUND 4,0,1,4,,,10
3320 DI:CALL &8000,aax,aay,spr(sp1) [1815]
3330 OUT &7F00,148                       [67]
3340 CALL &8000,ax,ay,spr(sp)           [2037]
3350 CALL &BD19:EI                       [500]
3360 sp1=sp:aax=ax:aay=ay               [1346]
3370 CALL &80A9,ax+tm,ay:IF PEEK(&A [3765]
011)=&CF THEN GOSUB 4550
3380 CALL &8032,ax,ay:IF PEEK(&A000 [4555]
)=0 THEN SOUND 2,80,200,4,,5:GOTO 3710
3390 IF ax=sx AND ay=sy AND sc1>=sc  [946]
 THEN 3740
3400 IF PEEK(&A010) THEN 3980           [1473]
3410 GOTO 3260                           [461]
3420 IF ax<bx THEN p=2:bx=bx-2:GOTO [2230]
 3440
3430 IF ax>bx THEN p=0:bx=bx+2          [671]
3440 IF ay<by THEN by=by-3:GOTO 3460[1121]
3450 IF ay>by THEN by=by+3              [1193]
3460 CALL &8000,bbx,bby,spr(pp1)        [1577]
3470 OUT &7F00,148                       [67]
3480 CALL &8000,bx,by,spr(pp)           [1200]
3490 CALL &8055,ax,bx,ay,by             [1569]
3500 bbx=bx:bby=by:pp1=pp                [625]
3510 ji=ji XOR 1:pp=p+ji                [1147]
3520 RETURN                              [555]
3530 SOUND 2,400,40,0,2,4               [1461]
3540 FOR i=ay TO ay+45 STEP 5           [1297]
3550 IF i>199 THEN 3620                 [1151]
3560 CALL &BD19                          [352]
3570 CALL &8000,aax,aay,spr(sp1)        [1203]
3580 CALL &8000,ax,i,spr(8)             [1325]
3590 CALL &BD19                          [352]
3600 ay=i:aay=ay:sp1=8                  [1015]
3610 NEXT                                [350]
3620 CALL &8032,ax,ay                    [807]
3630 IF PEEK(&A000)<>0 THEN SOUND 1 [4075]
30,0,1,5,,,31:sp1=8:GOTO 3380
3640 ay=ay-5                             [686]
3650 OUT &7F00,148                       [67]
3660 CALL &8000,aax,aay,spr(8)          [1377]
3670 CALL &8000,ax,ay,spr(8)            [1792]
3680 CALL &BD19                          [352]
3690 aax=ax:aay=ay                       [712]
3700 GOTO 3620                           [676]
3710 sp=8                                [188]
3720 CALL &8000,aax,aay,spr(sp1)        [1203]
3730 ay=ay-5:GOTO 3670                   [854]
3740 CALL &8000,aax,aay,spr(sp1)        [1203]
3750 CALL &8000,ax,ay,spr(8)            [1792]
3760 RETURN                              [555]
3770 FOR z=1 TO 1200:NEXT               [1836]
3780 MODE 0:PEN 8                       [1120]
3790 LOCATE 5,20:PRINT"SCORE : ";       [2254]
3800 PEN 9:PRINT USING"###";sco         [1786]
3810 PEN 6:a$="CIRCUIT"                 [1273]
3820 FOR i=1 TO 7                        [447]
3830 FOR j=2 TO 12                       [805]
3840 SOUND 1,j*40,1,5                   [1539]
3850 LOCATE 6+i,j:PRINT MID$(a$,i,1 [1184]
)
3860 LOCATE 6+i,j-1:PRINT" "             [677]
3870 CALL &BD19                          [352]
3880 NEXT:NEXT                          [1022]
3890 PEN tabl                            [515]
3900 FOR i=19 TO 15 STEP-1              [1613]
3910 SOUND 1,i*4-20,1,5                 [1601]
3920 LOCATE i,12:PRINT HEX$(tabl)"  [2221]
"
3930 FOR z=1 TO 100:NEXT                [1697]
3940 NEXT                                [350]
3950 PEN 1                               [549]
3960 FOR z=1 TO 3000:NEXT               [1109]
3970 MODE 0:RETURN                      [1315]
3980 '********************               [665]
3990 '*     c'est  perdu     *           [963]
4000 '********************               [665]
4010 dx=ax:dy=ay                         [604]
4020 SOUND 4,207,0,0,13,13              [1394]
4030 IF ax<80 THEN ax=150:GOTO 4050 [1387]
4040 ax=0                                 [85]
4050 IF ay<100 THEN ay=199:GOTO 407 [1032]
0
4060 ay=40                               [250]
4070 FOR i=1 TO 50                       [531]
4080 SOUND 1,INT(RND*400)+600,2,3,, [2400]
,INT(RND*10)+1
4090 INK 5,INT(RND*26)+1                [1550]
4100 INK 7,INT(RND*26)+1                 [911]
4110 INK 9,INT(RND*26)+1                [1245]
4120 INK 13,INT(RND*26)+1                [777]
4130 FOR z=1 TO 10:NEXT:NEXT            [1462]
4140 GOSUB 1370                          [921]
4150 CALL &8000,dx,dy,spr(sp1)          [1766]
4160 SOUND 1,800,80,0,2,,20             [1706]
4170 CALL &8000,dx,dy,spr(11)           [1474]
4180 FOR z=1 TO 200:NEXT                [2080]
4190 CALL &8000,dx,dy,spr(11)           [1474]
4200 CALL &8000,dx,dy,spr(12)           [1476]
4210 FOR z=1 TO 200:NEXT                [2080]
4220 CALL &8000,dx,dy,spr(12)           [1476]
4230 CALL &8000,dx,dy,spr(11)           [1474]
4240 FOR z=1 TO 200:NEXT                [2080]
4250 CALL &8000,dx,dy,spr(11)           [1474]
4260 FOR z=1 TO 500:NEXT                [1912]
4270 SOUND 4,1200,0,0,14,14             [1503]
4280 SOUND 4,0,20,2,,,10                [1358]
4290 FOR i=1000 TO 1500 STEP 100        [1698]
4300 SOUND 4,i,30,0,15,15               [1741]
4310 NEXT:z=REMAIN(0)                   [1505]
4320 WINDOW 7,14,9,13:PEN 11:PAPER  [1957]
3:CLS
4330 SYMBOL 255,6,8,60,102,126,96,6 [1758]
0
4340 LOCATE 3,2:PRINT"T'es"             [1530]
4350 LOCATE 2,4:PRINT"crev"CHR$(255 [1855]
)"!"
4360 PLOT 192,192,2:DRAWR 0,78:DRAW [1601]
R 252,0
4370 DRAWR 0,-78:DRAWR -252,0           [1213]
4380 FOR z=1 TO 6000:NEXT               [1418]
```

```
4390 CALL &8000,bbx,bby,spr(pp1)      [1577]
4400 CLS:PEN 1:LOCATE 2,2:PRINT"SCO  [1119]
RE:"
4410 LOCATE 3,4:PRINT USING"###";sc  [1928]
o
4420 PLOT 192,192,2:DRAWR 0,78:DRAW  [1601]
R 252,0
4430 DRAWR 0,-78:DRAWR -252,0        [1213]
4440 CALL &8000,bbx,bby,spr(pp1)     [1577]
4450 WHILE INKEY$<>"":WEND:EVERY 4   [3684]
GOSUB 3420
4460 ax=120:ay=87                    [531]
4470 WHILE ax<>bx OR ay<>by:IF INKE  [3915]
Y$<>"" THEN RUN 120 ELSE WEND
4480 ax=120:ay=159                   [534]
4490 WHILE ax<>bx OR ay<>by:IF INKE  [3915]
Y$<>"" THEN RUN 120 ELSE WEND
4500 ax=32:ay=159                    [584]
4510 WHILE ax<>bx OR ay<>by:IF INKE  [3915]
Y$<>"" THEN RUN 120 ELSE WEND
4520 ax=32:ay=87                     [945]
4530 WHILE ax<>bx OR ay<>by:IF INKE  [3915]
Y$<>"" THEN RUN 120 ELSE WEND
4540 GOTO 4460                       [317]
4550 DI:sc1=sc1+1                    [507]
4560 IF tm=-2 THEN tm=-8             [1407]
4570 CALL &8000,ax+tm,ay,spr(10)     [1126]
4580 SOUND 1,800,80,0,2,,31          [1707]
4590 CALL &8000,ax+tm,ay,spr(11)     [1128]
4600 CALL &BD19:CALL &BD19           [1099]
4610 CALL &8000,ax+tm,ay,spr(11)     [1128]
4620 CALL &8000,ax+tm,ay,spr(12)     [1122]
4630 CALL &BD19:CALL &BD19           [1099]
4640 CALL &8000,ax+tm,ay,spr(12)     [1122]
4650 CALL &8000,ax+tm,ay,spr(11)     [1128]
4660 CALL &BD19:CALL &BD19           [1099]
4670 CALL &8000,ax+tm,ay,spr(11)     [1128]
4680 sco=sco+10                      [562]
4690 IF sc=sc1 THEN CALL &8000,kx1,  [3181]
ky1,spr(9):CALL &8000,kx,ky,spr(9)
4700 EI:RETURN                       [416]
4710 '*******************            [665]
4720 '*     c'est gagne     *        [1461]
4730 '*******************            [665]
4740 FOR z=1 TO 800:NEXT:z=REMAIN(0  [2174]
)
4750 MODE 0:BORDER 0:GOSUB 1360      [1154]
4760 PEN 5:LOCATE 6,9:PRINT CHR$(24  [1667]
2);
4770 PEN 11:PAPER 5:PRINT" Z80"CHR$  [2905]
(149)"RAM ";
4780 PEN 5:PAPER 0:PRINT CHR$(243)   [2813]
4790 PEN 11:FOR i=10 TO 14           [1077]
4800 LOCATE 11,i:PRINT CHR$(149)     [1074]
4810 NEXT                            [350]
4820 CALL &8000,150,128,spr(9)       [1658]
4830 PAPER 15:PRINT CHR$(20):PAPER   [2085]
0
4840 ax=150:ay=103:x=600:y=224:z=48  [2762]
:GOSUB 2070
4850 k1=6:k=8                        [489]
4860 FOR i=150 TO 20 STEP-2          [994]
4870 OUT &7F00,148                   [67]
4880 CALL &8000,i,ay,spr(k)          [832]
4890 CALL &8000,i-2,ay,spr(k1)       [998]
4900 CALL &BD19                      [352]
4910 SOUND 4,0,1,4,,,10              [1116]
4920 k=k1                            [288]
4930 k1=k1 XOR 1                     [525]
4940 FOR z=1 TO 60:NEXT:NEXT         [1375]
4950 CALL &8000,i,ay,spr(k)          [832]
4960 CALL &8000,18,ay,spr(8)         [1792]
4970 SYMBOL 255,6,8,60,102,126,96,6  [1758]
0
4980 WINDOW 7,14,18,22:PAPER 4:CLS   [1374]
4990 PEN 1:LOCATE 3,2:PRINT"T'as"    [1875]
5000 LOCATE 2,4:PRINT"gagn"CHR$(255  [2655]
)"!"
5010 PLOT 192,48:DRAWR 252,0:DRAWR   [1509]
0,78
5020 DRAWR -252,0:DRAWR 0,-78        [916]
5030 WINDOW 1,20,1,8:PAPER 0         [1522]
5040 r$="F"+CHR$(255)+"licitations,  [9757]
 vous m'avez d"+CHR$(255)+"barass"+
CHR$(255)+" de ces sales bestioles.
"+SPACE$(10)+"Allez en paix, mon fi
ls !"+SPACE$(10)
5050 WHILE INKEY$<>"":WEND           [1786]
5060 FOR i=1 TO LEN(r$)              [755]
5070 LOCATE 20,4:PRINT MID$(r$,i,1)  [1762]
5080 FOR j=1 TO 4                    [621]
5090 OUT &7F00,148                   [67]
5100 CALL &BD19                      [352]
5110 CALL &80C2                      [492]
5120 IF INKEY$<>"" THEN RUN          [739]
5130 IF j=4 THEN 5150                [784]
5140 FOR z=1 TO 50:NEXT              [1522]
5150 NEXT j,i                        [396]
5160 GOTO 5060                       [343]
```

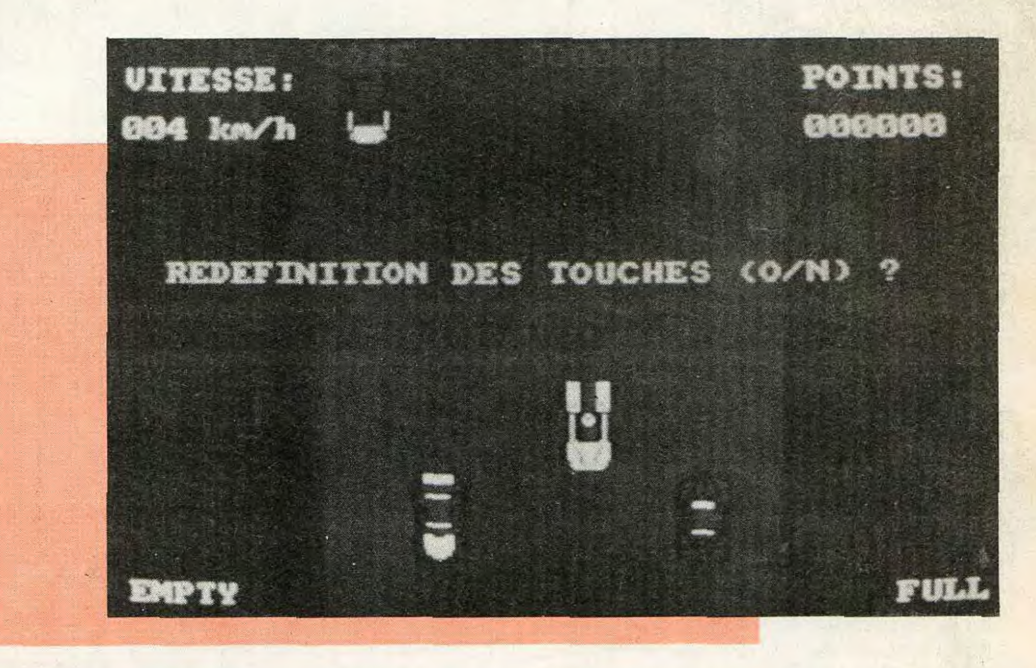


COCCINELLE

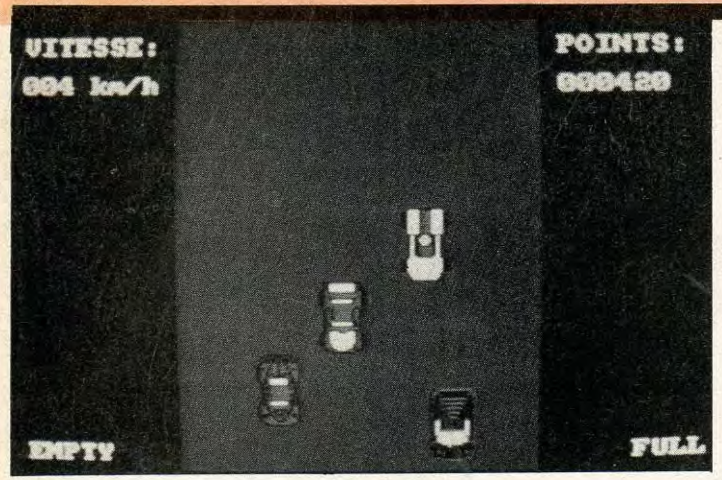
Au volant d'une brave coccinelle, éprouvez votre habileté en affrontant au mépris du danger, l'hostilité d'une circulation des plus intenses.
En évitant les véhicules et les plaques de verglas, songez à vous réapprovisionner en roulant sur les barils qui jalonnent le parcours. De temps à autre, un camion klaxonnant tentera de vous doubler. Laissez-lui le passage, sinon... Vous pouvez accélérer jusqu'à 250 km/h (joystick vers le haut ou flèche haut) et décélérer jusqu'à 50 km/h (joystick vers le bas ou flèche bas). A 250 km/h, 'Fire' enclenche le turbo...
Sauvegardez le listing Basic sous un nom de votre choix et entrez les codes hexadécimaux du second listing à l'aide du programme AMSAISIE (reportez-vous à son mode d'emploi). Spécifiez 82DC comme adresse de début et sauvez le langage machine par l'option 'S' sous le nom COC (à la suite du programme Basic dans le cas d'une sauvegarde K7). Au vu du nombre de codes, il semble raisonnable de morceler le travail en créant plusieurs fichiers (COC1, COC2, COC3, etc.), qui seront ultérieurement chargés à la suite après un MEMORY &82DB et sauvegardés ainsi dans un fichier unique :
SAVE''COC.BIN'',b,&82DC,&1C1F

Frédéric Carbonero

Nota : les possesseurs de 464 devront, aux lignes 190 et 210 du programme Basic, remplacer la fonction CLEAR INPUT par :
`WHILE INKEY$<>"":WEND`




```
10 MEMORY 32499:LOAD"COC.BIN",&82DC [3321]
:MODE 1:PEN 1:PAPER 0
20 RESTORE:INK 0,26:INK 1,0:INK 2,6 [1265]
30 BORDER 26:ENT -5,4,1,1,4,-1,1    [1915]
40 ENT -8,60,-1,1,60,1,1:ENV 1,2,34 [2106]
,50
50 LOCATE 10,10                     [676]
60 PRINT "Indications pour jouer (O [2430]
/N)"
70 IF INKEY(46)<>-1 THEN GOTO 200   [2044]
80 IF INKEY(34)=-1 THEN GOTO 70     [466]
90 CLS:PEN 2                        [491]
100 LOCATE 12,1:PRINT "INDICATIONS" [899]
110 LOCATE 12,2:PRINT "-----------" [1468]
120 PEN 1:LOCATE 5,4                [879]
130 PRINT ".Que vous redefinissiez  [12180]
les ":LOCATE 6,5:PRINT "touches ou
non, vous pouvez ":LOCATE 6,6:PRINT
"jouer au JOYSTICK.":LOCATE 5,8:PR
INT ".Si vous ne redefinissez pas l
es "
140 LOCATE 6,9:PRINT "touches, il y [7071]
 a 4 touches pre-":LOCATE 6,10:PRIN
T "programmees ";:LOCATE 21,11:PRIN
T "( ";
150 FOR f=240 TO 243:PRINT CHR$(f)+ [4102]
CHR$(32);:NEXT:PRINT ")":LOCATE 5,1
3
160 PRINT ".Vous pouvez accelerer j [10111]
usqu'a":LOCATE 6,14:PRINT "250 Km/h
.":LOCATE 5,16:PRINT ".Vous pouvez
decelerer jusqu'a":LOCATE 6,17:PRIN
T "50 Km/h."
170 LOCATE 5,19:PRINT ".A 250 Km/h  [4565]
vous pouvez activer "
180 LOCATE 6,20:PRINT "le TURBO qui [13071]
 vous propulsera a":LOCATE 6,21:PRI
NT "300 Km/h.":LOCATE 5,23:PRINT ".
```




```
Pour enlever le TURBO, il vous":LOC
ATE 6,24:PRINT "suffit de decelerer
."
190 CLEAR INPUT:CALL &BB18            [1145]
200 RESTORE:INK 0,0:INK 1,26:BORDER   [1717]
 0
210 CLEAR INPUT:GOSUB 330:z=1:MOVE    [4028]
230,352:GOSUB 250:MOVE 220,570:z=0.
8
220 GOSUB 250:z=2:MOVE 230,300        [1375]
230 GOSUB 250:z=4:MOVE 230,250        [2303]
240 GOSUB 250:GOTO  210               [1327]
250 READ a:IF a=255 THEN 310          [1145]
260 IF a=254 THEN 290                 [1122]
270 READ b:a=a/z:b=b/z:DRAWR a,b      [2343]
280 GOTO 250                          [419]
290 READ a,b:a=a/z:b=b/z:MOVER a,b    [1938]

300 GOTO 250                          [419]
310 RESTORE:FOR f=1 TO 96:a$=INKEY$   [3566]

320 IF a$<>"" THEN CALL &82DC  ELSE   [2101]
 NEXT f
330 MODE 2:LOCATE 50,25:PRINT "Prog   [5394]
ramme de F.CARBONERO":RETURN
340 DATA 0,-4,-10,-10,-16,0,-6,6,0,   [1527]
10,8
350 DATA 8,20,0,8,-8,6,-26,20,-20,1   [1690]
6,0
360 DATA 14,14,2,16,-8,8,-2,0,-2,2,   [1003]
-16
370 DATA 0,-4,-4,-2,0,-12,-12,-2,-1   [1936]
0,0
380 DATA -30,8,-50,-4,-10,-6,-6,-8,   [1358]
-2
390 DATA -20,0,-12,4,-6,6,-2,8,0,10   [1113]
,2,4
400 DATA 8,8,22,0,16,-14,4,-20,8,-8   [1189]
,8
410 DATA -2,20,4,14,12,0,12,0         [727]
420 DATA -20,8,-8,8,-2,14,0,14,14,4   [1573]
,12
430 DATA 0,-12,2,-2,4,-10,14,0,254,   [1722]
-20
440 DATA 20,0,22,-4,4,-2,0,-2,2,-8,   [1435]
2
450 DATA -10,0,-8,-4,-6,-6,-2,-6,-4   [801]
,-4
460 DATA -2,-4,254,-208,-82           [829]
470 DATA -16,-8,-14,0,-6,6,0,4,4,12   [1037]
,8,8
480 DATA 14,0,12,-4,6,-6,6,-14,6,-6   [1556]
,8
490 DATA -4,18,0,20,6,6,6,2,10,-4,1   [1984]
0,-8
500 DATA 4,-14,0,-12,-6,-30,-30,-18   [1356]
,-46
510 DATA -4,-16,0,-28,4,-14,14,-14,   [1939]
14
520 DATA -6,14,0,22,4,10,10,4,10,-4   [867]
,18
530 DATA -8,8,-8,2,-6,0,-10,-4,-4,-   [1721]
4,0
540 DATA -12,6,-6,6,0,2,2             [1178]
550 DATA 254,24,-12,6,0,2,2,4,0,0,2   [1165]
,6
560 DATA -14,6,-2,12,0,12,4,10,10,2   [1424]
,4,2
570 DATA 12,0,18,-2,8,-10,10,-10,0,   [1241]
-16
580 DATA -6,-4,-4,-6,-22,-2,-16,254   [1592]
,32
590 DATA 48,2,-10,4,-4,22,0,0,2,10,   [1127]
10
600 DATA 14,0,2,-2,254,-4,-50,0,-4,   [1109]
-6
610 DATA -6,-12,0,-6,6,-4,12,0,18,4   [2158]
,12
620 DATA 254,60,12,-2,2,-14,0,-10,-   [1218]
10
630 DATA -6,-14,0,-18,4,-14,6,-6,12   [1086]
,0,6
640 DATA 6,0,4,254,0,-10,10,2,10,10   [1522]
,12
650 DATA 26,4,20,254,0,8,2,0,2,2,0,   [1279]
2,-2
660 DATA 2,-2,0,-2,-2,0,-2,254,-4,-   [1282]
34,0
670 DATA -14,-2,-2,0,-8,2,-2,0,-4,4   [1762]
,-4
680 DATA 8,0,2,2,22,58,2,2,8,0,2,-2   [1339]
,0
690 DATA -12,-2,-2,-8,-44,254,12,54   [1775]
,6,6
700 DATA 6,0,4,-4,0,-20,-2,-2,-2,-1   [1926]
6,2
710 DATA -12,4,-4,6,0,10,10           [881]
720 DATA 0,16,8,24,12,12,12,0,2,-2,   [2024]
0,-2
730 DATA 2,-2,0,-6,-2,-8,-6,-6,-14,   [1191]
0,-6
740 DATA 6,254,-6,-34,8,-8,8,0,8,2,   [701]
34
750 DATA 34,0,2,10,10,30,58,0,4,-2,   [1803]
8,-4
760 DATA 4,-8,0,-10,-10,-4,-10,-2,-   [1218]
8,-2
770 DATA -36,0,-50,2,-2,0,-2,4,-4,1   [1745]
0,0
780 DATA 2,2,2,0,6,6,8,38,4,2         [797]
790 DATA 30,58,0,4,-2,8,-4,4,-8,0,-   [820]
10
800 DATA -10,-4,-10,-2,-8,-2,-36,0,   [1142]
-50
810 DATA 2,-2,0,-2,4,-4,10,0,10,10,   [1190]
0,16
820 DATA 8,24,12,12,12,0,2,-2,0,-2,   [626]
2,-2
830 DATA 0,-6,-2,-8,-6,-6,-14,0,-6,   [1718]
6
840 DATA 254,-6,-34,8,-8,8,0,8,2,25   [3291]
4,40,-2,2,0,2,2,0,2,-2,2,-2,0,-2,-2
,0,-2,255
```




```
82DC:3E 01 32 E6 9E 3E 08 32:CB
82E4:E7 9E 3E 00 32 E8 9E 3E:1F
82EC:02 32 E9 9E 3E 2F 32 EA:B2
82F4:9E 21 EB 9E 06 FF 36 00:F9
82FC:23 10 FB 3E 00 06 00 0E:FE
8304:00 CD 32 BC 3E 01 06 00:87
830C:0E 00 CD 32 BC 3E 02 06:9E
8314:06 0E 06 CD 32 BC 3E 03:AD
831C:06 1A 0E 1A CD 32 BC 3E:E0
8324:01 CD 0E BC 3E 00 CD 96:E0
832C:BB 3E 02 CD 90 BB 21 01:E4
8334:01 CD 75 BB 21 CD 9E 06:47
833C:13 7E CD 5A BB 23 10 F9:5E
8344:06 06 7E 0E 30 81 CD 5A:37
834C:BB 23 10 F6 06 03 21 0A:E7
8354:04 CD 75 BB 3E 03 CD 90:76
835C:BB 21 9A 9D 06 20 7E CD:63
8364:5A BB 23 10 F9 CD 18 BB:C8
836C:3E 2E CD 1E BB C2 6B 84:B2
8374:3E 22 CD 1E BB 28 EE 3E:51
837C:01 CD 0E BC 3E 02 CD 90:34
8384:BB 3E 00 CD 96 BB 21 07:46
838C:05 CD 75 BB 21 BA 9D 06:8F
8394:0F 7E CD 5A BB 23 10 F9:B2
839C:3E 03 CD 90 BB CD 18 BB:18
83A4:CD 5A BB CD 58 84 32 E6:CA
83AC:9E 3E 02 CD 90 BB 21 09:4F
83B4:05 CD 75 BB 21 C9 9D 06:C6
83BC:0F 7E CD 5A BB 23 10 F9:DA
83C4:3E 03 CD 90 BB CD 18 BB:40
83CC:CD 5A BB CD 58 84 32 E7:F3
83D4:9E 3E 02 CD 90 BB 21 0B:79
83DC:05 CD 75 BB 21 D8 9D 06:FD
83E4:0F 7E CD 5A BB 23 10 F9:02
83EC:3E 03 CD 90 BB CD 18 BB:68
83F4:CD 5A BB CD 58 84 32 E8:1C
83FC:9E 3E 02 CD 90 BB 21 0D:A3
8404:05 CD 75 BB 21 E7 9D 06:35
840C:0F 7E CD 5A BB 23 10 F9:2B
8414:3E 03 CD 90 BB CD 18 BB:91
841C:CD 5A BB CD 58 84 32 E9:46
8424:9E 3E 02 CD 90 BB 21 0F:CE
842C:05 CD 75 BB 21 3D 9E 06:B4
8434:0F 7E CD 5A BB 23 10 F9:53
843C:3E 03 CD 90 BB CD 18 BB:B9
8444:CD 5A BB CD 58 84 32 EA:6F
844C:9E 21 FF FF 2B 7C FE 00:32
8454:20 FA 18 13 3E 00 06 48:A9
845C:F5 C5 CD 1E BB 20 05 C1:26
8464:F1 3C 10 F4 C1 F1 C9 3E:D2
846C:00 CD 0E BC 21 0B 0B CD:BB
8474:75 BB 21 1F 9E 06 06 7E:90
847C:CD 5A BB 23 10 F9 21 0D:3C
8484:01 CD 75 BB 21 25 9E 06:F0
848C:0F 7E CD 5A BB 23 10 F9:AB
8494:21 0F 0B CD 75 BB 21 34:A5
849C:9E 06 09 7E CD 5A BB 23:50
84A4:10 F9 3E 35 CD 1E BB 20:6A
84AC:1C 3E 26 CD 1E BB 20 0E:84
84B4:3E 3D CD 1E BB 28 EB 3E:AA
84BC:03 32 1C 9F 18 0C 3E 02:94
84C4:32 1C 9F 18 05 3E 01 32:C3
84CC:1C 9F 3E 01 CD 0E BC 3E:1F
84D4:15 32 EE 9E 3E 37 32 EF:C1
84DC:9E 21 CC E7 22 F2 9E 3E:C2
84E4:26 32 ED 9E 21 C5 C5 22:18
84EC:EB 9E 3E 32 32 F0 9E 21:4A
84F4:01 01 CD 75 BB 3E 01 CD:83
84FC:90 BB 2A FA 9E 2E 00 22:DD
8504:FA 9E 3E 00 32 17 9F 32:79
850C:18 9F 32 19 9F 32 1A 9F:1D
8514:3E 00 06 00 0E 00 CD 32:EA
851C:BC 3E 01 06 00 0E 00 CD:7D
8524:32 BC 3E 02 06 00 0E 00:EB
852C:CD 32 BC 3E 03 06 00 0E:C1
8534:00 CD 32 BC 3E 00 CD 96:15
853C:BB 3E 01 CD 90 BB 06 19:F2
8544:48 06 0A C5 3E 8F CD 5A:DA
854C:BB C1 10 F7 06 14 C5 3E:71
8554:80 CD 5A BB C1 10 F7 06:09
855C:0A C5 3E 8F CD 5A BB C1:20
8564:10 F7 41 10 DB 3E 00 06:60
856C:01 0E 01 CD 32 BC 3E 01:FB
8574:06 00 0E 00 CD 32 BC 3E:06
857C:02 06 06 0E 06 CD 32 BC:DE
8584:3E 03 06 1A 0E 1A CD 32:91
858C:BC 2A EB 9E DD 21 B9 92:C9
8594:CD 97 92 3E 01 CD 96 BB:6C
859C:3E 03 CD 90 BB 21 02 02:9F
85A4:CD 75 BB 21 F6 9D 06 08:EB
85AC:7E CD 5A BB 23 10 F9 21:DE
85B4:04 02 CD 75 BB 21 FE 9D:F8
85BC:06 08 7E CD 5A BB 23 10:E2
85C4:F9 21 02 20 CD 75 BB 21:A3
85CC:06 9E 06 07 7E CD 5A BB:62
85D4:23 10 F9 21 04 20 CD 75:0C
85DC:BB 21 F9 9E 06 05 7E 0E:6B
85E4:30 81 CD 5A BB 2B 10 F6:2D
85EC:3E 30 CD 5A BB 21 18 02:FC
85F4:CD 75 BB 21 0D 9E 06 05:4D
85FC:7E CD 5A BB 23 10 F9 21:2E
8604:18 24 CD 75 BB 21 12 9E:94
860C:06 04 7E CD 5A BB 23 10:2F
8614:F9 21 94 E7 0E 04 06 28:6F
861C:36 F0 23 10 FB 0D 28 06:31
8624:11 D8 07 19 18 F0 CD 2F:B7
862C:86 18 1F 2A F2 9E 7D FE:A4
8634:80 D8 06 7F 98 21 80 E7:B7
863C:0E 04 E5 47 36 0F 23 10:78
8644:FB E1 0D C8 11 00 08 19:AD
864C:18 F0 06 08 21 00 9F 36:DE
8654:FF 23 10 FB 06 04 21 FC:2E
865C:9E 36 00 23 10 FB 2A FA:08
8664:9E 23 22 FA 9E C3 AE 89:5F
866C:7C FE 01 30 05 7D FE 32:4F
8674:38 7E 3A FC 9E FE 01 28:AB
867C:43 ED 5F E6 0C FE 00 20:A1
8684:6F 2A 02 9F 7C FE C0 38:B6
868C:70 2A 04 9F 7C FE C0 38:C1
8694:68 2A 06 9F 7C FE C0 38:C3
869C:60 2A FA 9E 7D 21 D6 BE:76
86A4:E6 1F 06 17 80 32 0E 9F:AB
86AC:98 85 6F 22 00 9F 3E 01:BE
86B4:32 FC 9E DD 21 59 93 CD:BD
86BC:0F 91 18 3D 2A 00 9F 11:11
86C4:50 00 19 22 00 9F 7C FE:EE
86CC:C7 38 09 FE C8 30 0E 7D:DB
86D4:FE 30 30 09 DD 21 59 93:AB
86DC:CD 0F 91 18 1C 3E 00 32:73
86E4:FC 9E 11 50 00 ED 52 DD:81
86EC:21 5D 9B CD 0F 91 18 09:19
86F4:21 F4 01 2B 7C FE 00 20:55
86FC:FA 7C FE 01 30 05 7D FE:A7
8704:3C 38 7F 3A FD 9E FE 01:52
870C:28 44 ED 5F E6 0C FE 00:3B
8714:20 70 2A 00 9F 7C FE C0:2E
871C:38 71 2A 04 9F 7C FE C0:53
8724:38 69 2A 06 9F 7C FE C0:55
872C:38 61 2A FA 9E 7D 85 21:31
8734:D6 BE E6 1F 06 17 80 32:23
873C:0F 9F 98 85 6F 22 02 9F:C0
8744:3E 01 32 FD 9E DD 21 F9:CE
874C:93 CD 0F 91 18 3D 2A 02:54
8754:9F 11 50 00 19 22 02 9F:B7
875C:7C FE C7 38 09 FE C8 30:5B
8764:0E 7D FE 30 30 09 DD 21:DB
876C:F9 93 CD 0F 91 18 1C 3E:5E
8774:00 32 FD 9E 11 50 00 ED:16
877C:52 DD 21 5D 9B CD 0F 91:B8
8784:18 09 21 F4 01 2B 7C FE:E7
878C:00 20 FA 3A 1C 9F FE 01:21
8794:CA CA 88 7C FE 01 30 06:E8
879C:7D FE 46 DA 23 88 3A FE:A1
87A4:9E FE 01 28 46 ED 5F E6:68
87AC:0C FE 00 20 72 2A 00 9F:98
87B4:7C FE C0 38 73 2A 02 9F:EB
87BC:7C FE C0 38 6B 2A 06 9F:EF
87C4:7C FE C0 38 63 2A FA 9E:E2
87CC:7D 21 D6 BE 85 85 80 E6:F5
87D4:1F 06 17 80 32 10 9F 98:90
87DC:85 6F 22 04 9F 3E 01 32:8D
87E4:FE 9E DD 21 99 94 CD 0F:0E
87EC:91 18 3D 2A 04 9F 11 50:87
87F4:00 19 22 04 9F 7C FE C7:9A
87FC:38 09 FE C8 30 0E 7D FE:43
8804:30 30 09 DD 21 99 94 CD:ED
880C:0F 91 18 1C 3E 00 32 FE:D6
8814:9E 11 50 00 ED 52 DD 21:D8
881C:5D 9B CD 0F 91 18 09 21:4B
8824:F4 01 2B 7C FE 00 20 FA:60
882C:3A 1C 9F FE 02 CA CA 88:C5
8834:7C FE 01 30 06 7D FE 32:1A
883C:DA C1 88 3A FF 9E FE 01:BD
8844:28 47 ED 5F E6 0C FE 00:77
884C:20 73 2A 00 9F 7C FE C0:6A
8854:38 74 2A 02 9F 7C FE C0:8D
885C:38 6C 2A 04 9F 7C FE C0:8F
8864:38 64 2A FA 9E 7D 21 D6:BE
886C:BE 85 85 81 87 E6 1F 06:CF
8874:17 80 32 11 9F 98 85 6F:01
887C:22 06 9F 3E 01 32 FF 9E:D9
8884:DD 21 39 95 CD 0F 91 18:5D
888C:3D 2A 06 9F 11 50 00 19:9A
8894:22 06 9F 7C FE C7 38 09:65
889C:FE C8 30 0E 7D FE 30 30:03
88A4:09 DD 21 39 95 CD 0F 91:6E
88AC:18 1C 3E 00 32 FF 9E 11:86
88B4:50 00 ED 52 DD 21 5D 9B:C1
88BC:CD 0F 91 18 09 21 F4 01:E8
88C4:2B 7C FE 00 20 FA 3A 17:5C
88CC:9F FE 00 C2 68 8A 2A 00:CF
88D4:9F 7C FE C4 38 2D 20 05:C3
88DC:7D FE 60 38 26 7C FE C7:DE
88E4:30 21 FE C6 20 05 7D FE:21
88EC:8F 30 18 3A ED 9E 06 03:19
88F4:98 47 3A 0E 9F B8 38 0B:3D
88FC:3E 09 80 47 3A 0E 9F B8:31
8904:DA E0 8B 2A 02 9F 7C FE:17
890C:C4 38 2D 20 05 7D FE 60:8E
8914:38 26 7C FE C7 30 21 FE:8B
891C:C6 20 05 7D FE 8F 30 18:E2
8924:3A ED 9E 06 03 98 47 3A:94
892C:0F 9F B8 38 0B 3E 09 80:25
8934:47 3A 0F 9F B8 DA E0 8B:E9
893C:2A 04 9F 7C FE C4 38 2D:35
8944:20 05 7D FE 60 38 26 7C:A7
894C:FE C7 30 21 FE C6 20 05:D4
8954:7D FE 8F 30 18 3A ED 9E:F4
895C:06 03 98 47 3A 10 9F B8:6E
8964:38 0B 3E 09 80 47 3A 10:88
896C:9F B8 DA E0 8B 2A 06 9F:60
8974:7C FE C4 DA 68 8A 20 06:2D
897C:7D FE 60 DA 68 8A 7C FE:26
8984:C7 D2 68 8A FE C6 20 06:82
```

```
898C:7D FE 8F D2 68 8A 3A ED:0A
8994:9E 06 03 98 47 3A 11 9F:8D
899C:B8 DA 68 8A 3E 09 80 47:B7
89A4:3A 11 9F B8 DA E0 8B C3:D7
89AC:68 8A 3A 19 9F FE 01 28:40
89B4:37 2A FA 9E 7D E6 7F FE:16
89BC:3C C2 6C 86 2A FA 9E 3A:31
89C4:ED 9E 86 E6 1F 06 17 80:00
89CC:32 14 9F 98 21 16 C0 85:4E
89D4:6F 22 0A 9F 3E 01 32 19:21
89DC:9F DD 21 F5 9C CD 01 92:F3
89E4:3E 01 32 15 9F C3 6C 86:47
89EC:3A 15 9F 3C 32 15 9F FE:83
89F4:21 20 12 2A 0A 9F DD 21:A1
89FC:5D 9B CD 01 92 3E 00 32:4D
8A04:19 9F C3 6C 86 2A 0A 9F:CE
8A0C:11 00 30 19 7C FE 50 30:EA
8A14:04 11 50 C0 19 22 0A 9F:A7
```




```
8A1C:DD 21 F5 9C CD 01 92 3A:CF
8A24:15 9F FE 15 DA 6C 86 3A:7B
8A2C:17 9F FE 00 20 0A 2A EB:A9
8A34:9E DD 21 B9 92 CD 97 92:9B
8A3C:FE 1C D2 6C 86 3A ED 9E:69
8A44:06 02 98 47 3A 14 9F B8:5A
8A4C:DA 6C 86 3E 08 80 47 3A:E9
8A54:14 9F B8 D2 6C 86 21 88:B6
8A5C:9D CD AA BC 3E 0A 32 1B:4B
8A64:9F C3 6C 86 3A 18 9F FE:31
8A6C:01 28 35 2A FA 9E 7D FE:91
8A74:00 C2 38 8B 7C 80 84 CB:CE
8A7C:17 CB 17 E6 1F 06 17 80:A1
8A84:32 12 9F 98 21 16 C0 85:05
8A8C:6F 22 08 9F 3E 01 32 18:D7
8A94:9F DD 21 0D 9B CD E3 91:A4
8A9C:3E 01 32 13 9F C3 38 8B:CF
8AA4:3A 13 9F 3C 32 13 9F FE:38
8AAC:2C 20 11 2A 08 9F DD 21:62
8AB4:5D 9B CD E3 91 3E 00 32:E7
8ABC:18 9F 18 78 2A 08 9F 11:6F
8AC4:00 20 19 7C FE 50 30 04:85
8ACC:11 50 C0 19 22 08 9F DD:36
8AD4:21 0D 9B CD E3 91 3A 13:B5
8ADC:9F FE 21 38 57 3A ED 9E:78
8AE4:06 03 98 47 3A 12 9F B8:F9
8AEC:38 4A 3E 08 80 47 3A 12:51
8AF4:9F B8 30 40 3E 00 32 18:CD
8AFC:9F 2A 08 9F DD 21 5D 9B:EC
8B04:CD E3 91 3A 17 9F FE 00:BE
8B0C:20 0A 2A EB 9E DD 21 B9:2B
8B14:92 CD 97 92 21 7F 9D CD:31
8B1C:AA BC 2A F2 9E 06 05 7D:4F
8B24:FE CF 28 03 2C 10 F8 22:FD
8B2C:F2 9E 3A F7 9E 3C 32 F7:7B
8B34:9E CD 2F 86 3A 17 9F FE:CD
8B3C:00 C2 E0 8B 3A 1B 9F FE:E6
8B44:00 28 07 3D 32 1B 9F C3:EA
8B4C:E9 8C 3A E6 9E CD 1E BB:B0
8B54:20 1C 3A E7 9E CD 1E BB:80
8B5C:20 20 CD 24 BB 7C E6 08:3D
8B64:FE 08 28 23 7C E6 04 FE:A4
8B6C:04 28 28 C3 E9 8C 3A ED:AA
8B74:9E 47 3A EF 9E B8 30 28:BB
8B7C:18 E0 3A ED 9E 47 3A EE:33
8B84:9E B8 38 3A C3 E9 8C 3A:49
8B8C:ED 9E 47 3A EF 9E B8 30:98
8B94:0F 18 D1 3A ED 9E 47 3A:5D
8B9C:EE 9E B8 38 21 C3 E9 8C:FC
8BA4:2A EB 9E 3A ED 9E 3C 23:06
8BAC:47 3A EF 9E B8 38 02 04:3B
8BB4:23 78 32 ED 9E 22 EB 9E:42
8BBC:CD 4F 92 C3 E9 8C 2A EB:42
8BC4:9E 3A ED 9E 3D 2B 47 3A:9B
8BCC:EE 9E B8 30 02 05 2B 78:75
8BD4:32 ED 9E 22 EB 9E CD 71:05
8BDC:92 C3 E9 8C 3A F1 9E FE:F8
8BE4:00 20 08 3E 01 32 F1 9E:97
8BEC:C3 6E 8C 3E 00 32 F1 9E:33
8BF4:2A EB 9E 3A 17 9F 3C 32:90
8BFC:17 9F FE 07 CA C1 8C FE:57
8C04:09 CA D0 8C 2A EB 9E 3A:AC
8C0C:17 9F FE 01 28 68 FE 02:DD
8C14:CA 99 8C FE 03 CA A3 8C:89
8C1C:FE 04 CA B2 8C FE 05 CA:7F
8C24:B2 8C FE 06 CA B2 8C FE:F8
8C2C:07 28 74 FE 08 28 70 FE:F7
8C34:09 28 62 FE 0A 28 5E FE:DF
8C3C:0B 28 3B FE 0C 28 37 FE:9D
8C44:0D CA DF 8C FE 19 DA 6E:71
8C4C:8C 3E 00 32 17 9F 2A F2:A6
8C54:9E 06 14 2B 10 FD 22 F2:E4
8C5C:9E 3E 01 CD 0E BC 21 B8:35
8C64:0B 2B 7C FE 00 20 FA C3:7D
8C6C:E3 84 21 B8 0B 2B 7C FE:E8
8C74:00 20 FA C3 BA 8E DD 21:23
8C7C:D9 95 CD 97 92 3A F0 9E:34
8C84:CB 3F 47 3E 82 98 06 32:F1
8C8C:80 32 7D 9D 21 76 9D CD:E5
8C94:AA BC C3 6E 8C DD 21 79:BA
8C9C:96 CD 97 92 C3 6E 8C 11:82
8CA4:51 00 ED 52 DD 21 19 97:6E
8CAC:CD 6D 91 C3 6E 8C 11 52:23
8CB4:00 ED 52 DD 21 31 98 CD:13
8CBC:C1 91 C3 6E 8C 11 52 00:BA
8CC4:ED 52 DD 21 5D 9B CD C1:13
8CCC:91 C3 08 8C 11 51 00 ED:8F
8CD4:52 DD 21 5D 9B CD 6D 91:73
8CDC:C3 08 8C DD 21 5D 9B CD:82
8CE4:97 92 C3 6E 8C 3A E8 9E:16
8CEC:CD 1E BB 20 1B 3A E9 9E:1A
8CF4:CD 1E BB 20 20 CD 24 BB:12
8CFC:7C E6 01 FE 01 28 09 7C:97
8D04:E6 02 FE 02 28 0F 18 18:E0
8D0C:3A F0 9E FE FA 30 11 3C:D6
8D14:32 F0 9E 18 0B 3A F0 9E:4C
8D1C:FE 33 38 04 3D 32 F0 9E:13
8D24:3A 1D 9F FE 01 28 36 21:25
8D2C:04 02 CD 75 BB 21 F1 9E:6C
8D34:36 30 3A F0 9E FE 64 38:89
8D3C:06 06 64 98 34 18 F6 F5:08
8D44:7E CD 5A BB F1 36 30 FE:86
8D4C:0A 38 06 06 0A 98 34 18:15
8D54:F6 F5 7E CD 5A BB F1 06:23
8D5C:30 80 CD 5A BB 3A F4 9E:47
8D64:3C FE 64 28 05 32 F4 9E:80
8D6C:18 17 3E 00 32 F4 9E 2A:54
8D74:F2 9E 2B 22 F2 9E 23 06:97
8D7C:04 36 F0 11 00 08 19 10:75
8D84:F8 2A FA 9E 7D E6 07 FE:33
8D8C:00 C2 BA 8E 3A F0 9E FE:E9
8D94:FA 20 07 3A F6 9E 3C 32:7E
8D9C:F6 9E 3A 1D 9F FE 01 20:D2
8DA4:07 3A F7 9E 3C 32 F7 9E:0A
8DAC:3A F0 9E E6 C0 06 05 CB:7D
8DB4:3F 10 FC 47 3A F5 9E 80:20
8DBC:32 F5 9E FE 0A 38 0D 06:61
8DC4:0A 98 32 F5 9E 3A F6 9E:86
8DCC:3C 32 F6 9E 3A F6 9E FE:27
8DD4:0A 38 0D 06 0A 98 32 F6:80
8DDC:9E 3A F7 9E 3C 32 F7 9E:D9
8DE4:3A F7 9E FE 0A 38 0D 06:93
8DEC:0A 98 32 F7 9E 3A F8 9E:B2
8DF4:3C 32 F8 9E 3A F8 9E FE:53
8DFC:0A 38 4C 06 0A 98 32 F8:E9
8E04:9E 3A F9 9E 3C 32 F9 9E:06
8E0C:3A F8 9E 3C 32 F8 9E 21:8F
8E14:0A 0D CD 75 BB 21 A9 9E:1E
8E1C:06 0C 7E CD 5A BB 23 10:4F
8E24:F9 06 04 21 E8 FD 2B 7C:62
8E2C:FE 00 20 FA 10 F5 21 0A:02
8E34:0D CD 75 BB 3E 00 CD 90:67
8E3C:BB 06 0C 3E 8F CD 5A BB:46
8E44:10 F9 3E 03 CD 90 BB 3A:6E
8E4C:F9 9E FE 0A 38 1F 06 0A:E0
8E54:98 32 F9 9E 3E 00 CD 0E:5C
8E5C:BC 21 06 01 CD 75 BB 21:EC
8E64:5A 9E 06 4F 7E CD 5A BB:9F
8E6C:23 10 F9 18 FE DD 21 6D:A7
8E74:9D DD 36 07 01 21 6D 9D:E5
8E7C:CD AA BC DD 21 6D 9D DD:22
8E84:36 07 02 21 04 20 CD 75:D8
8E8C:BB 3A F9 9E 06 30 80 CD:29
8E94:5A BB 3A F8 9E 06 30 80:BD
8E9C:CD 5A BB 3A F7 9E 06 30:11
8EA4:80 CD 5A BB 3A F6 9E 06:68
8EAC:30 80 CD 5A BB 3A F5 9E:99
8EB4:06 30 80 CD 5A BB 3A 1A:2E
8EBC:9F FE 01 28 34 2A FA 9E:06
8EC4:7C E6 07 FE 07 C2 46 8F:57
8ECC:7D FE 64 20 06 21 91 9D:AE
8ED4:CD AA BC FE A0 20 6B 21:DF
8EDC:55 FF 22 0C 9F 3E 01 32:FC
8EE4:1A 9F DD 21 E1 99 CD 28:98
8EEC:92 3E 01 32 16 9F C3 46:3B
8EF4:8F 3A 16 9F 3C 3C 32 16:C0
8EFC:9F FE BD 20 12 2A 0C 9F:EB
8F04:DD 21 5D 9B CD 28 92 3E:4E
8F0C:00 32 1A 9F C3 46 8F 2A:48
8F14:0C 9F 11 FF 0F ED 52 7C:28
8F1C:FE C0 30 05 11 4F C0 ED:AB
8F24:52 22 0C 9F DD 21 E1 99:4A
8F2C:CD 28 92 3A 16 9F FE 10:3F
8F34:38 10 FE 62 30 0C 3A ED:CE
8F3C:9E FE 22 38 05 FE 2C DA:CA
8F44:E0 8B 3A F0 9E FE FA 20:1E
8F4C:56 3A 1D 9F FE 01 28 79:C7
8F54:3A EA 9E CD 1E BB 20 0C:77
8F5C:CD 24 BB 7C E6 10 FE 10:17
8F64:28 02 18 65 3E 01 32 1D:28
8F6C:9F 21 04 02 CD 75 BB 3E:FC
8F74:33 CD 5A BB 3E 30 CD 5A:AD
8F7C:BB 21 0B 22 CD 75 BB 21:32
8F84:4C 9E 06 05 7E CD 5A BB:68
8F8C:23 10 F9 21 0D 20 CD 75:D7
8F94:BB 21 51 9E 06 09 7E CD:48
8F9C:5A BB 23 10 F9 18 2A 3A:E8
8FA4:1D 9F FE 00 28 23 3E 00:76
8FAC:32 1D 9F 21 0B 22 CD 75:B9
8FB4:BB 06 05 3E 80 CD 5A BB:A9
8FBC:10 F9 21 0D 20 CD 75 BB:9F
8FC4:06 09 3E 80 CD 5A BB 10:12
8FCC:F9 2A F2 9E 7C FE E7 C2:31
8FD4:DE 90 7D FE 80 D2 DE 90:0C
8FDC:3A F0 9E 3D 3D 32 F0 9E:6D
8FE4:FE 02 28 09 FE 01 28 05:D0
8FEC:FE 00 C2 DE 90 06 03 21:D3
8FF4:FF FF 2B 7C FE 00 20 FA:40
8FFC:10 F5 3E 02 CD 90 BB 3E:26
9004:00 CD 96 BB 21 0A 10 CD:BA
900C:75 BB 21 16 9E 06 09 7E:2E
9014:CD 5A BB 23 11 50 C3 1B:E8
901C:7A FE 00 20 FA 10 F0 06:44
9024:05 21 FF FF 2B 7C FE 00:7D
```

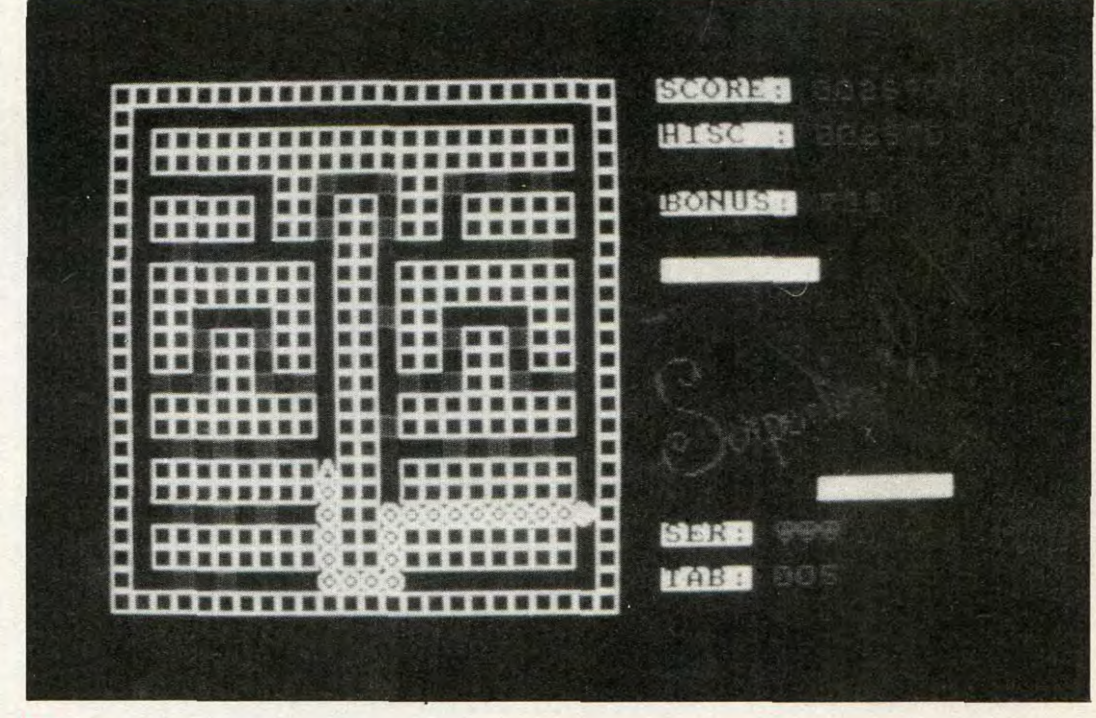
```
902C:20 FA 10 F5 3A E0 9E 47:DA
9034:3A F9 9E B8 DA F5 82 20:BE
903C:30 3A E1 9E 47 3A F8 9E:CC
9044:B8 DA F5 82 20 23 3A E2:3C
904C:9E 47 3A F7 9E B8 DA F5:17
9054:82 20 16 3A E3 9E 47 3A:D8
905C:F6 9E DA F5 82 20 0A 3A:35
9064:F5 9E 47 3A E4 9E D2 F5:51
906C:82 3A F5 9E 32 E4 9E 3A:39
9074:F6 9E 32 E3 9E 3A F7 9E:1A
907C:32 E2 9E 3A F8 9E 32 E1:A1
9084:9E 3A F9 9E 32 E0 9E 3E:71
908C:00 06 00 0E 00 CD 32 BC:EB
9094:3E 03 06 1A 0E 06 CD 32:98
909C:BC 3E 03 CD 90 BB 3E 01:80
90A4:CD 0E BC 21 05 01 CD 75:34
90AC:BB 21 B5 9E 06 28 7E CD:E4
90B4:5A BB 23 10 F9 21 08 01:AF
90BC:CD 75 BB 21 E0 9E 06 06:F4
90C4:7E 0E 30 81 CD 5A BB 23:96
90CC:10 F6 06 0A 21 EB FD 2B:A3
90D4:7C FE 00 20 FA 10 F5 C3:C0
90DC:F5 82 3A F0 9E CB 3F CB:80
90E4:3F 47 3E FF 98 32 70 9D:0E
90EC:21 6D 9D CD AA BC 3A F0:04
90F4:9E 47 3E FC 98 26 00 6F:D0
90FC:06 05 29 10 FD 2B 7C FE:72
9104:00 20 FA 7D FE 00 20 F5:3F
910C:C3 62 86 E5 7C FE C0 38:9F
9114:13 0E 08 06 05 3E 00 77:8E
911C:23 10 FA 0D 28 06 11 FB:21
9124:07 19 18 EF E1 11 50 00:1E
912C:19 16 04 D5 E5 0E 08 7C:3C
9134:FE C0 30 07 11 28 00 DD:D0
913C:19 18 14 06 05 DD 7E 00:78
9144:77 23 DD 23 10 F7 0D 28:AB
914C:06 11 FB 07 19 18 EC E1:F4
9154:11 50 00 19 7C FE C8 30:D1
915C:0E FE C7 38 05 7D FE 80:FB
9164:30 05 D1 15 C8 18 C4 E1:95
916C:C9 E5 11 00 20 19 0E 04:07
9174:06 07 DD 7E 00 77 23 DD:E4
917C:23 10 F7 11 F9 07 19 0D:6E
9184:20 EE E1 11 50 00 19 16:94
918C:04 D5 E5 0E 08 06 07 DD:DB
9194:7E 00 77 23 DD 23 10 F7:44
919C:11 F9 07 19 0D 20 EE E1:53
91A4:11 50 00 19 D1 15 20 E1:96
91AC:0E 04 06 07 DD 7E 00 77:2E
91B4:DD 23 23 10 F7 11 F9 07:80
91BC:19 0D 20 EE C9 16 06 D5:3B
91C4:E5 0E 08 06 09 DD 7E 00:BA
91CC:77 23 DD 23 10 F7 11 F7:06
91D4:07 19 0D 20 EE E1 11 50:E2
91DC:00 19 D1 15 C8 18 E0 0E:3A
91E4:14 06 04 DD 7E 00 77 23:88
91EC:DD 23 10 F7 11 FC 07 19:B1
91F4:0D C8 7C FE 80 30 EA 11:7F
91FC:50 C0 19 18 E4 0E 1E 06:E4
9204:04 DD 7E 00 77 23 DD 23:8F
920C:10 F7 11 FC 07 19 0D C8:A7
9214:7C FE 80 30 EA 11 50 C0:DB
921C:19 7C FE C7 38 E1 7D FE:9C
9224:80 38 DC C9 0E 34 06 06:61
922C:DD 7E 00 77 23 DD 23 10:C3
9234:F7 11 FA 07 19 0D C8 7C:39
923C:FE 80 30 EA 11 50 C0 19:A0
9244:7C FE C7 38 E1 7D FE 80:2B
924C:38 DC C9 DD 21 B9 92 E5:E9
9254:2B 2B E5 0E 04 06 08 3E:7F
925C:00 77 23 77 11 FF 07 19:2F
9264:10 F5 E1 11 50 00 19 0D:63
926C:28 28 E5 18 E8 DD 21 B9:EA
9274:92 E5 23 23 23 23 23 E5:11
927C:E1 04 06 08 3E 00 77 23:06
9284:77 11 FF 07 19 10 F5 E1:A3
928C:11 50 00 19 0D 28 03 E5:B5
9294:18 EB E1 16 04 D5 E5 0E:E9
929C:08 06 05 DD 7E 00 77 23:36
92A4:DD 23 10 F7 11 FB 07 19:69
92AC:0D 20 EE E1 11 50 00 19:B4
92B4:D1 15 C8 18 E0 00 F0 F0:CC
92BC:E0 00 30 00 E0 10 80 40:0E
92C4:F0 5A E0 40 30 5A 5A 5A:FE
92CC:80 43 2D 4B 87 4B 87 4B:3A
92D4:4B 4B 2C A5 4B 4B 4B A4:52
92DC:87 87 4B 2D 2C 87 A5 4B:97
92E4:A5 2C 87 96 F0 2D 2C 96:43
92EC:3C F0 96 2C 96 7B FF DA:56
92F4:2C 52 7B FF DA 48 21 78:39
92FC:F0 C3 80 21 87 0F 2D 80:25
9304:21 87 0F 2D 80 21 0F 0F:3A
930C:0F 80 21 87 0F 2D 80 21:B3
9314:87 0F 2D 80 52 0F 0F 1E:78
931C:48 52 78 F0 D2 48 52 7B:98
9324:FF DA 48 52 3C F0 96 48:34
932C:87 87 0F 2D 2C 87 96 F0:42
9334:2D 2C 87 5A 1E 4B 2C B4:4A
933C:4B A5 5A A4 B4 2D 4B 96:7F
9344:A4 43 1E 1E 0F 48 43 0F:A3
934C:E1 0F 48 30 0F 0F 1E 80:03
9354:00 F0 F0 E0 00 03 0F 0F:C8
935C:0F 0C 12 78 F0 E1 84 12:FB
9364:78 F0 E1 84 16 78 F0 E1:23
936C:86 3C 0F 0F 0F C3 3C F0:DD
9374:F0 F0 C3 2D F0 F0 F0 4B:F2
937C:3C 0F 0F 0F C3 3C B7 FF:2D
9384:DE C3 3C C3 0F 3C C3 3C:01
938C:F0 F0 F0 C3 25 78 F0 E1:20
9394:4A 16 B4 F0 D2 86 12 B4:49
939C:F0 D2 84 12 B4 F0 D2 84:81
93A4:12 B4 F0 D2 84 12 B4 F0:F9
93AC:D2 84 12 87 0F 1E 84 12:F1
93B4:3F FF CF 84 12 0F 0F 0F:17
93BC:84 12 F0 F0 F0 84 12 0F:5A
93C4:F0 0F 84 16 78 0F E1 86:DE
93CC:3C 78 F0 E1 C3 3C 78 F0:4B
93D4:E1 C3 3C 78 F0 E1 C3 3C:8F
93DC:78 F0 E1 C3 3C B4 F0 D2:2D
93E4:C3 3C B4 F0 D2 C3 12 87:48
93EC:F0 1E 84 03 78 0F E1 0C:88
93F4:00 07 0F 0E 00 30 F0 F0:BB
93FC:F0 C0 21 F7 FF FE 48 21:BD
9404:F7 FF FE 48 61 F7 FF FE:29
940C:68 C3 F0 F0 F0 3C C3 0F:A9
9414:0F 0F 3C D2 0F 0F 0F B4:B5
941C:C3 F0 F0 F0 3C C3 7B FF:BC
9424:ED 3C C3 3C F0 C3 3C C3:92
942C:0F 0F 0F 3C 52 87 0F 1E:2F
9434:A4 61 4B 0F 2D 68 21 4B:28
943C:0F 2D 48 21 4B 0F 2D 48:44
9444:21 4B 0F 2D 48 21 4B 0F:43
944C:2D 48 21 78 F0 E1 48 21:28
9454:F3 FF FC 48 21 F0 F0 F0:0F
945C:48 21 0F 0F 0F 48 21 F0:DF
9464:0F F0 48 61 F7 F0 FE 68:ED
946C:C3 F7 FF FE 3C C3 F7 FF:AC
9474:FE 3C C3 F7 FF FE 3C C3:F8
947C:F7 FF FE 3C C3 7B FF ED:6A
9484:3C C3 7B FF ED 3C 21 78:53
948C:FF E1 48 30 87 F0 1E C0:CD
9494:00 78 F0 E1 00 30 F0 F0:81
949C:F0 C0 31 FE 0F F7 C8 F1:CE
94A4:FE 0F F7 F8 F1 FE 0F F7:29
94AC:F8 F1 FE 0F F7 F8 F1 FE:14
94B4:0F F7 F8 F1 FE 0F F7 F8:33
94BC:F1 FE 0F F7 F8 F1 FA 0F:37
94C4:F5 F8 31 F6 0F F6 C8 31:6A
94CC:FE 0F F7 C8 10 F0 69 F0:85
94D4:80 10 F8 F6 F1 80 10 F8:5F
94DC:F9 F1 80 10 F8 FF F1 80:52
94E4:10 F8 F6 F1 80 10 E9 69:49
94EC:79 80 10 E9 0F 79 80 10:8A
94F4:E9 0F 79 80 10 E9 0F 79:FA
94FC:80 10 F8 0F F1 80 30 FC:C4
9504:F0 F3 C0 F1 FF FF FF F8:22
950C:F1 FB FF FD F8 F1 FD FF:6E
9514:FB F8 F1 FD FF FB F8 F1:6D
951C:FE FF F7 F8 F1 FE FF F7:82
9524:F8 10 FE FF F7 80 10 FF:44
952C:F6 FF 80 00 F7 FF FE 00:2A
9534:00 70 F0 E0 00 03 0F 0F:2A
953C:0F 0C 07 F0 F0 F0 0E 1E:EF
9544:F0 F0 F0 87 3C F0 F0 F0:3C
954C:C3 78 0F 0F 0F E1 3C F0:56
9554:F0 F0 C3 78 87 0F 1E E1:99
955C:78 F0 F0 F0 E1 69 C3 0F:55
9564:3C 69 5A F0 F0 F0 A5 69:D6
956C:E1 0F 78 69 5A F0 F0 F0:FC
9574:A5 78 F0 0F F0 E1 69 78:D7
957C:F0 E1 69 5B 78 F0 E1 AD:9C
9584:5B 78 F0 E1 AD 5B 78 F0:2D
958C:E1 AD 5B 78 F0 E1 AD 5B:5B
9594:0F 0F 0F AD 5B 7F FF EF:CB
959C:AD 4B 7F FF EF 2D 4B FF:0D
95A4:FF FF 2D 4B FF FF FF 2D:D9
95AC:0F 3F FF CF 0F 3C 87 0F:3E
95B4:1E C3 78 B4 F0 D2 E1 78:71
95BC:B4 F0 D2 E1 25 5A F0 A5:BC
95C4:4A 25 5A F0 A5 4A 34 D2:07
95CC:F0 B4 C2 03 5A F0 A5 0C:C5
95D4:00 07 0F 0E 00 00 00 00:8D
95DC:00 00 00 00 00 00 00 00:71
95E4:00 00 00 00 00 00 00 00:79
95EC:00 00 70 10 D0 00 00 F7:C8
95F4:B1 FD 80 10 FF F9 FE C8:85
95FC:10 FF FD F6 C8 10 FF FD:67
9604:F6 C8 11 FB FB FD 88 11:F5
960C:F7 FE FA 88 10 F9 FF FF:20
9614:80 10 F7 FF FF 80 00 F6:A5
961C:FC FF 80 00 F6 FF F6 00:18
9624:00 F3 F7 E4 00 10 FD FF:94
962C:F6 00 10 FB FE FF 80 31:71
9634:F7 FD FB 80 31 F7 FF FD:5D
963C:88 10 FB F3 FB 88 11 FA:E6
9644:FD FD 88 11 FF FE FD 80:E7
964C:10 F7 F3 FD 80 31 FF FD:86
9654:FD 80 31 FF FF FA 00 10:A0
965C:FF F7 E4 00 00 F0 70 40:6C
9664:00 00 00 00 00 00 00 00:FA
966C:00 00 00 00 00 00 00 00:02
9674:00 00 00 00 00 00 E0 10:FA
967C:D0 C0 10 FE 31 FD EC 31:FB
9684:FF F3 FE FE 73 FF FB FE:73
968C:FE 73 F3 FB F6 FE 73 FD:E5
9694:FB F6 FE 73 FF FB F6 FE:7A
969C:76 FF F7 FE EC 76 FF F7:F4
96A4:FD C0 75 FF FC FB 80 73:55
96AC:FF FF FF C8 73 FF FF FF:77
96B4:EC 73 F1 F8 FF EC 70 FF:EC
96BC:F7 F7 EC 00 F7 F7 F7 C0:D1
96C4:10 FF FF F4 00 31 FF FF:8B
96CC:F7 80 73 FB FE FF C8 F7:03
96D4:F7 FD FF EC F6 FF FB FF:38
96DC:E4 F6 FF FF FE EC F6 FF:29
96E4:FF FF E4 F7 F7 F0 FE EE:26
96EC:F7 FE FF F7 E6 73 FF FF:C4
96F4:FB F6 70 F7 FE FD F6 F7:CA
96FC:FF FF F1 FF F6 F7 FE FF:6A
9704:F6 F7 FF FF FF F6 73 FF:ED
970C:FF FE EC 31 FF FB FD C8:7C
9714:10 F0 B0 F0 80 00 F0 C0:7B
971C:30 F0 F0 C0 10 F0 E0 F0:53
9724:F0 80 E0 30 FF FC F3 FF:28
972C:F3 F8 71 FF FE F7 FF F9:0B
9734:F8 71 FF FF F7 FF FD FC:21
973C:71 FF FF F7 FF FD FC 71:A2
9744:FF FF F7 FB FD F4 71 FF:2C
974C:FF F7 FB FD F4 71 F9 FF:2E
9754:F7 FB FD F4 10 F1 FF F7:C5
975C:FB FD C0 10 F7 FE FF F7:A6
9764:FA 00 30 FF FD FE FF F7:15
976C:00 71 FF FF FF FF FC 00:6C
9774:71 FC F7 FF FF FE 80 71:5C
977C:F8 F7 FF FC FF C0 71 F3:20
9784:FE F1 FF F7 C0 71 F3 FE:22
978C:F0 FF F7 C0 71 F3 FF FE:2A
```

```
9794:F7 F7 C0 31 F3 FF FF F2:ED
979C:FF C0 31 F3 FF FF F3 FF:06
97A4:C0 30 F9 FF FF F3 FE 80:93
97AC:30 FC F1 FF F3 FC 00 30:7E
97B4:FE F1 FE F7 E8 00 30 FF:46
97BC:FF FC FF F8 00 30 FE FF:72
97C4:F9 FF FC 80 00 F7 F0 F3:A9
97CC:FF FE C0 00 73 F8 F6 F3:74
97D4:FF C0 10 F1 FF FF F9 FF:21
97DC:C0 30 F1 FF FF FC FF A0:ED
97E4:71 FF FF FF FC FE B0 F3:B6
97EC:FE F0 FF FC FE FC F3 FE:57
97F4:F0 F7 FC F9 FC F3 FF FF:54
97FC:F3 FD F9 FC F3 FF FF F9:62
9804:FB F9 FC F3 FF FF FD FF:79
980C:F9 FC F3 FF FF FF FF F9:B1
9814:FC F1 FF FE FF FF F9 E8:75
981C:70 FF FC F7 FF F1 C0 30:F6
9824:F0 E0 F0 F0 B0 B0 10 F0:9C
982C:C0 70 F0 30 00 00 30 F0:34
9834:F0 00 70 F0 E0 00 00 70:6C
983C:F0 F0 80 F0 F0 F0 00 00:04
9844:F3 FF FE D0 F3 F9 FC 80:04
984C:10 F7 FF FF F0 F7 FC FE:CA
9854:C0 10 F7 FF FF F8 FD FE:A4
985C:F7 C0 10 F7 FF FF FC FE:AA
9864:FE F7 C0 10 F7 FF FB FC:AE
986C:FE FE F7 C0 10 F7 FF F7:B4
9874:FC FE FE F7 C0 10 F7 FE:C0
987C:F7 FC FE FE F7 C0 10 F6:C0
9884:FC FF FC FE FE F7 C0 10:D6
988C:FC FE FF FE FD FE F7 C0:CB
9894:11 F9 FC FF FF FF FC FE:29
989C:80 11 F3 FC FF FF FF F8:A9
98A4:F6 00 10 F7 FE F7 FE F7:23
98AC:F3 F3 00 10 F7 FF F0 F3:13
98B4:F3 FF F9 00 10 F7 FF F8:35
98BC:F1 FB FF FC 00 10 F6 F0:31
98C4:FF FF FB FF FE 80 10 F4:D6
98CC:F1 FF FF FB FF FE 80 10:DB
98D4:F0 FF FF FF F3 F3 FE 80:BD
98DC:10 F1 FE F1 FE F6 F7 FE:4D
98E4:80 10 F3 FC F3 FC FC FF:E5
98EC:FE 80 30 F7 F8 FF F9 FD:16
98F4:FF FE 80 30 FF F9 FF F3:23
98FC:F9 FF FE 80 30 FF F9 F8:2A
9904:F7 FB FF FE 80 30 FF F9:34
990C:FF FF F3 FF FC 00 30 FF:C0
9914:F9 FF FE F7 FF E8 00 30:B1
991C:FF F9 FF FC FF F8 C0 00:5F
9924:30 FF F9 FF FC FF F0 E0:AF
992C:00 30 F7 FC FF FC FF FF:E1
9934:FC 00 10 F3 FE F3 FC FF:B8
993C:FF FE 80 00 F0 FF F1 FC:2E
9944:FF F3 FF 80 10 F8 FF FF:54
994C:FC FF F9 FF 80 31 FF FF:87
9954:FF FC FF FC FF 80 31 FF:92
995C:FF FF FC FF FE F7 80 31:94
9964:FC F0 F3 FC FF FF F3 80:49
996C:10 FB F0 F1 FE F7 FF F3:D5
9974:C0 00 F0 FF FC FE F1 FF:A6
997C:F3 C8 10 F1 FF FE FF F8:C5
9984:F7 F3 C8 30 F3 FF FF FF:EF
998C:FE F3 F3 C8 30 F7 FF FF:F6
9994:FF FF FF F3 C8 30 F7 FF:0B
999C:F0 F0 F7 FF F3 C8 30 F7:ED
99A4:FE FF FF FB FF F3 C8 30:1E
99AC:F7 FD FF FF FD FF F3 C8:EE
99B4:30 F3 FF FF F3 FE FE F3:50
99BC:C8 10 F1 FF FC F1 FF FC:05
99C4:F7 C0 00 F0 FF F8 70 FF:6A
99CC:F8 FE 80 00 70 F0 E0 30:4B
99D4:F0 F1 FC 00 00 30 F0 C0:2A
99DC:10 F0 F0 E0 00 00 00 F0:35
99E4:F0 00 00 F0 70 0F 0F E0:CB
99EC:F0 87 C3 0F 0F 3C 1E 87:BE
99F4:C3 0F 0F 3C 1E 87 87 0F:E5
99FC:0F 1E 1E 87 B4 F0 F0 D2:CD
9A04:1E 96 7B FE F7 ED 96 96:DB
9A0C:7B FE F7 ED 96 52 79 FE:62
9A14:F7 E9 A4 30 78 F0 F0 E1:9B
9A1C:C0 10 69 0F 0F 69 80 30:26
9A24:69 0F 0F 69 C0 30 69 0F:16
9A2C:0F 69 C0 30 69 0F 0F 69:1E
9A34:C0 30 69 0F 0F 69 C0 30:9E
9A3C:F0 F0 F0 F0 C0 00 F0 F0:36
9A44:F0 F0 00 70 F0 F0 F0 F0:EE
9A4C:E0 73 FF FF FF FF EC 73:94
9A54:FF FF FF FF EC 73 FF FF:47
9A5C:FF FF EC 73 FF FF FF FF:4F
9A64:EC 73 FF FF FF FF EC 73:B8
9A6C:FF FF FF FF EC 73 FF FF:5F
9A74:FF FF EC 73 FF FF FF FF:67
9A7C:EC 73 FF FF FF FF EC 71:CE
9A84:FF FF FF FF E8 72 F0 F0:54
9A8C:F0 F0 E4 71 FF FF FF FF:57
9A94:E8 73 FF FF FF FF EC 73:E4
9A9C:FF FF FF FF EC 73 FF FF:8F
9AA4:FF FF EC 73 FF FF FF FF:97
9AAC:EC 73 FF FF FF FF EC 73:00
9AB4:FF FF FF FF EC 73 FF FF:A7
9ABC:FF FF EC 73 FF FF FF FF:AF
9AC4:EC 73 FF FF FF FF EC 71:16
9ACC:FF FF FF FF E8 72 F0 F0:9C
9AD4:F0 F0 E4 71 FF FF FF FF:9F
9ADC:E8 73 FF FF FF FF EC 73:2C
9AE4:FF FF FF FF EC 73 FF FF:D7
9AEC:FF FF EC 73 FF FF FF FF:DF
9AF4:EC 73 FF FF FF FF EC 73:48
9AFC:FF FF FF FF EC 73 FF FF:EF
9B04:FF FF EC 30 F0 F0 F0 F0:79
9B0C:C0 00 00 00 00 00 00 00:67
9B14:00 00 00 00 00 00 00 00:AF
9B1C:00 00 70 E0 00 10 F0 F0:F7
9B24:80 30 F0 F0 C0 70 F0 F0:5F
9B2C:E0 70 F0 F0 E0 F0 1E 5A:3F
9B34:F0 F0 5A 5A F0 F0 5A 5A:F7
9B3C:F0 F0 5A 5A F0 F0 5A 5A:FF
9B44:F0 F0 1E 4B 78 70 F0 F0:F0
9B4C:E0 70 F0 F0 E0 30 F0 F0:07
9B54:C0 10 F0 F0 80 00 70 E0:6F
9B5C:00 00 00 00 00 00 00 00:F7
9B64:00 00 00 00 00 00 00 00:FF
9B6C:00 00 00 00 00 00 00 00:07
9B74:00 00 00 00 00 00 00 00:0F
9B7C:00 00 00 00 00 00 00 00:17
9B84:00 00 00 00 00 00 00 00:1F
9B8C:00 00 00 00 00 00 00 00:27
9B94:00 00 00 00 00 00 00 00:2F
9B9C:00 00 00 00 00 00 00 00:37
9BA4:00 00 00 00 00 00 00 00:3F
9BAC:00 00 00 00 00 00 00 00:47
9BB4:00 00 00 00 00 00 00 00:4F
9BBC:00 00 00 00 00 00 00 00:57
9BC4:00 00 00 00 00 00 00 00:5F
9BCC:00 00 00 00 00 00 00 00:67
9BD4:00 00 00 00 00 00 00 00:6F
9BDC:00 00 00 00 00 00 00 00:77
9BE4:00 00 00 00 00 00 00 00:7F
9BEC:00 00 00 00 00 00 00 00:87
9BF4:00 00 00 00 00 00 00 00:8F
9BFC:00 00 00 00 00 00 00 00:97
9C04:00 00 00 00 00 00 00 00:A0
9C0C:00 00 00 00 00 00 00 00:A8
9C14:00 00 00 00 00 00 00 00:B0
9C1C:00 00 00 00 00 00 00 00:B8
9C24:00 00 00 00 00 00 00 00:C0
9C2C:00 00 00 00 00 00 00 00:C8
9C34:00 00 00 00 00 00 00 00:D0
9C3C:00 00 00 00 00 00 00 00:D8
9C44:00 00 00 00 00 00 00 00:E0
9C4C:00 00 00 00 00 00 00 00:E8
9C54:00 00 00 00 00 00 00 00:F0
9C5C:00 00 00 00 00 00 00 00:F8
9C64:00 00 00 00 00 00 00 00:00
9C6C:00 00 00 00 00 00 00 00:08
9C74:00 00 00 00 00 00 00 00:10
9C7C:00 00 00 00 00 00 00 00:18
9C84:00 00 00 00 00 00 00 00:20
9C8C:00 00 00 00 00 00 00 00:28
9C94:00 00 00 00 00 00 00 00:30
9C9C:00 00 00 00 00 00 00 00:38
9CA4:00 00 00 00 00 00 00 00:40
9CAC:00 00 00 00 00 00 00 00:48
9CB4:00 00 00 00 00 00 00 00:50
9CBC:00 00 00 00 00 00 00 00:58
9CC4:00 00 00 00 00 00 00 00:60
9CCC:00 00 00 00 00 00 00 00:68
9CD4:00 00 00 00 00 00 00 00:70
9CDC:00 00 00 00 00 00 00 00:78
9CE4:00 00 00 00 00 00 00 00:80
9CEC:00 00 00 00 00 00 00 00:88
9CF4:00 00 00 00 00 00 00 00:90
9CFC:00 00 00 00 00 00 00 00:98
9D04:00 00 00 00 00 00 00 00:A1
9D0C:00 22 11 00 88 77 55 77:A7
9D14:CC 77 FF FF EE 77 FF FF:55
9D1C:CC 77 DD FF 88 33 FF FF:91
9D24:88 33 FF FF DD 11 FF FF:66
9D2C:FF 55 FF FF EE 77 FF 99:18
9D34:EE 33 FF 99 FF 00 FF 33:BB
9D3C:FF 77 FF FF FF 77 FF FF:C1
9D44:11 FF FF FF 11 11 FF FF:0F
9D4C:CC 44 FF FF CC 44 FF FF:05
9D54:EE 11 FF FF FF 33 FF FF:1E
9D5C:77 77 FF EE 22 22 FF DD:F4
9D64:66 00 EE DD 66 11 88 44:75
9D6C:00 01 00 00 96 00 00 08:A8
9D74:02 00 04 00 01 90 01 1F:C8
9D7C:0F 00 00 02 00 0B 64 00:96
9D84:00 0C 78 00 02 00 05 8E:3A
9D8C:00 00 0C 1E 00 04 01 00:58
9D94:F4 01 00 0B C8 00 52 45:90
9D9C:44 45 46 49 4E 49 54 49:85
9DA4:4F 4E 20 44 45 53 20 54:4E
9DAC:4F 55 43 48 45 53 20 28:58
9DB4:4F 2F 4E 29 20 3F 44 20:09
9DBC:72 20 6F 20 69 20 74 20:97
9DC4:65 20 20 20 3A 20 47 20:61:28
9DCC:20 75 20 63 20 68 20 65:8E
9DD4:20 20 3A 20 41 63 63 65:77
9DDC:6C 65 72 61 74 69 6F 6E:D7
9DE4:20 3A 20 44 65 63 65 6C:D8
9DEC:65 72 61 74 69 6F 6E 20:9B
9DF4:3A 20 56 49 54 45 53 53:C9
9DFC:45 3A 30 35 30 20 6B 6D:A5
9E04:2F 68 50 4F 49 4E 54 53:16
9E0C:3A 45 4D 50 54 59 46 55:0E
9E14:4C 4C 47 41 4D 45 20 4F:D3
9E1C:56 45 52 46 61 63 69 6C:86
9E24:65 4E 49 56 45 41 55 3A:29
9E2C:20 20 20 4D 6F 79 65 6E:32
9E34:44 69 66 66 69 63 69 6C:EC
9E3C:65 74 20 75 20 72 20 62:5C
9E44:20 6F 20 20 20 20 3A 20:4B
9E4C:54 55 52 42 4F 45 4E 43:4C
9E54:4C 45 4E 43 48 45 42 52:35
9E5C:41 56 4F 20 21 21 21 20:83
9E64:20 20 20 20 20 56 4F 55:9C
9E6C:53 20 41 56 45 5A 20 52:25
9E74:45 4D 50 4C 49 20 56 4F:4E
9E7C:54 52 45 20 43 4F 4E 54:59
9E84:52 41 54 20 2E 20 20 4C:E3
9E8C:27 4F 52 44 49 4E 41 54:62
9E94:45 55 52 20 56 4F 55 53:8B
9E9C:20 46 45 4C 49 43 49 54:5A
9EA4:45 20 21 21 21 42 4F 4E:E9
9EAC:55 53 20 20 31 30 30 30:F3
9EB4:30 42 52 41 56 4F 20 2C:48
9EBC:20 56 4F 55 53 20 44 45:70
9EC4:54 45 4E 45 5A 20 4C 45:99
9ECC:20 4D 45 49 4C 4C 45 55:97
9ED4:52 20 53 43 4F 52 45 20:80
9EDC:20 3A 20 20 00 00 00 00:14
9EE4:00 00 00 00 00 00 00 00:82
9EEC:00 00 00 00 00 00 00 00:8A
9EF4:00 00 00 00 00 00 00 00:92
```

SERPENTINO !

Gageons que l'efficience du langage machine appliquée à un thème aussi classique que celui du Nibbler (serpent express à croissance ultime), suscitera l'enthousiasme que Serpentino mérite.
Chemin faisant dans moult labyrinthes, un gai reptile doit gober en un temps limite, tous les objets favorisant sa croissance en s'efforçant de ne pas déguster son propre corps. La rapidité et la vitesse d'accélération sont réglables en début de jeu et un serpent supplémentaire vous est offert tous *les quatre tableaux. Le bonus* restant, le score et la vitesse augmentent à chaque étape.
Sauvez tout d'abord (sur disquette ou au début d'une cassette) le premier programme basic. Lancez ensuite le second programme disposant de ses propres vérificateurs, afin de créer et sauvegarder automatiquement les fichiers binaires : **DESSIN.BIN, TABLE.BIN, MUSIQ.BIN, SERP 1.BIN, SERP2.BIN** et **SERP3. BIN** qui constituent le programme principal.

Patrice Erieau





```
10 INK 0,0:BORDER 0                      [620]
20 MEMORY &50FF                          [415]
30 GOTO 440                              [444]
40 FOR i=&7D00 TO &7DFF STEP 2:POKE [2460]
 i,1:POKE i+1,0:NEXT
50 LOAD"!table",&6000                    [1546]
60 LOAD"!musiq",&7A00                    [948]
70 LOAD"!serp1",&9000                    [515]
80 LOAD"!serp2",&9400                    [645]
90 LOAD"!serp3",&8000                    [526]
100 MODE 1                               [506]
110 INK 0,0:INK 1,8:INK 2,24:INK 3, [1722]
18
120 PEN 3:LOCATE 1,10:PRINT"Rapidit [3139]
e du jeu :";
130 PEN 2:LOCATE 25,10:PRINT " - "; [1640]
140 PEN 1:FOR k=1 TO 7:PRINT CHR$(2 [2432]
33);:NEXT
150 PEN 2:PRINT" +"                      [705]
160 i=31:y=10                            [535]
170 PEN 2:LOCATE i,y:PRINT CHR$(143 [2959]
)
180 FOR k=1 TO 100:NEXT                  [2108]
190 IF JOY(0)=8 AND i<34 THEN PEN 1 [2917]
:LOCATE i,y:PRINT CHR$(233):i=i+1:G
OTO 170
200 IF JOY(0)=4 AND i>28 THEN PEN 1 [4153]
:LOCATE i,y:PRINT CHR$(233):i=i-1:G
OTO 170
210 IF JOY(0)<16 THEN 170                [1688]
220 PEN 3:LOCATE 1,17:PRINT"Vitesse [3905]
 d'acceleration :";
230 PEN 2:PRINT" - ";                    [935]
240 PEN 1:FOR k=1 TO 4:PRINT CHR$(2 [2225]
33);:NEXT
250 PEN 2:PRINT" +"                      [705]
260 x=29:y=17                            [1195]
```

SERPENTINO !

```
270 PEN 2:LOCATE x,y:PRINT CHR$(143 [1687]
)
280 FOR k=1 TO 100:NEXT            [2108]
290 IF JOY(0)=8 AND x<31 THEN PEN 1 [2609]
:LOCATE x,y:PRINT CHR$(233):x=x+1:G
OTO 270
300 IF JOY(0)=4 AND x>28 THEN PEN 1 [4666]
:LOCATE x,y:PRINT CHR$(233):x=x-1:G
OTO 270
310 IF JOY(0)<16 THEN 270          [1660]
320 RESTORE 350                    [741]
330 FOR t=28 TO i                  [1553]
340 READ a:NEXT                    [717]
350 DATA 241,201,161,121,81,41,1   [1186]
360 RESTORE 390                    [733]
370 FOR t=28 TO x                  [1602]
380 READ b:NEXT                    [685]
390 DATA 5,10,20,40                [700]
400 MODE 1:INK 0,0:BORDER 0:INK 1,1 [3519]
4:INK 2,24:INK 3,6
410 POKE &9291,a:POKE &91C2,b:POKE  [2951]
&928D,50
420 POKE &7F27,a:POKE &7F25,50:CALL [1889]
 &8000:END
430 '--dessin--                    [922]
440 MODE 0:LOAD"!dessin",&5A00:INK  [2390]
3,0:INK 4,0
450 FOR u=&5A00 TO &5F18 STEP 2    [1718]
460 PLOT (PEEK(u)*2)+100,PEEK(u+1)* [2378]
2+80,4
470 PLOT (PEEK(u)*2)+104,PEEK(u+1)* [2467]
2+82,3
480 NEXT                           [350]
490 SPEED INK 25,25:INK 3,8,0:INK 4 [2500]
,18
500 SYMBOL 240,24,60,102,126,255,21 [2782]
9,219,126
510 SYMBOL 241,60,102,195,153,153,1 [1997]
95,102,60
520 SYMBOL 242,126,195,195,102,36,6 [2451]
0,24,24
530 SYMBOL 243,126,219,219,255,126, [1919]
102,60,24
540 SYMBOL 244,24,24,60,36,102,195, [2266]
195,126
550 SYMBOL 245,14,63,121,223,223,12 [2154]
1,63,14
560 SYMBOL 246,96,240,156,135,135,1 [1756]
56,240,96
570 SYMBOL 247,112,248,158,251,251, [2127]
158,248,112
580 SYMBOL 248,6,15,57,225,225,57,1 [2009]
5,6
590 SYMBOL 249,0,24,24,126,126,24,2 [2338]
4,0
600 SYMBOL 255,255,129,129,129,129, [2182]
129,129,255
610 SYMBOL 250,0,66,36,24,24,36,66, [2095]
0
620 SYMBOL 251,24,126,126,255,255,1 [2488]
26,126,24
630 SYMBOL 252,240,144,240,143,136, [2578]
142,8,15
640 SYMBOL 253,255,213,171,213,171, [2268]
213,171,255
650 SYMBOL 254,0,126,66,90,90,66,12 [1844]
6,0
660 GOTO 40                        [378]
```

```
10 MODE 2
20 PRINT:PRINT:PRINT:PRINT
30 PRINT"    1 minute ........!!!"
40 PRINT:PRINT"Je charge tous les codes mac
hine....":PRINT
50 '
60 '
70 RESTORE 180
80 MEMORY &59FF
90 compt=0:lig=180
100 FOR i=&5A00 TO &5F19
110 READ a$
120 a=VAL("&"+a$):IF a<256 THEN 140
130 IF compt<>a THEN 2610 ELSE i=i-1:GOSUB
2630:compt=0:lig=lig+10:GOTO 150
140 POKE i,a:compt=compt+a
150 NEXT
160 READ a$:a=VAL("&"+a$):IF a<>compt THEN
2610 ELSE GOSUB 2630
170 PRINT:PRINT"ok":PRINT
180 DATA 1C,4E,1A,4E,1A,4C,1C,4C,1C,4A,1E,4
A,20,4A,22,4A,24,4C,24,4E,26,4E,28,50,26,50
,24,50,28,52,26,52,24,52,28,54,26,54,24,54,
26,56,24,56,26,58,24,58,24,5A,22,5A,20,5C,1
E,5C,1C,5C,1A,5C,18,5C,16,5C,14,5A,12,3E,10
,3E,10,40,0E,40,0E,42,1080
190 DATA 0C,42,0E,44,0C,44,0A,46,0A,46,0C,4
6,0E,46,0E,48,0C,48,0A,48,0E,4A,0C,4A,0A,4A
,08,4C,0A,4C,0C,4C,0C,4E,0A,4E,08,4E,08,50,
0A,50,0C,50,0C,52,0A,52,0C,54,0E,56,10,58,1
2,5A,16,5C,14,3C,16,3A,18,38,1A,38,1C,38,1E
,36,20,36,22,36,24,36,d02
200 DATA 26,34,28,34,2A,34,2C,32,2E,32,2E,3
0,30,30,32,30,32,2E,34,2E,36,16,38,16,3A,18
,38,18,3A,1A,38,1A,3A,1C,38,1C,38,1E,3A,1E,
3A,20,38,20,38,22,36,24,38,24,3A,22,38,26,3
6,26,38,28,36,28,36,2A,34,28,34,2A,36,2C,34
,2C,1E,08,20,08,22,08,cd8
210 DATA 24,08,26,08,26,0A,28,0A,2A,0A,2C,0
C,2E,0C,2E,0E,30,0E,32,0E,32,10,34,10,34,12
,36,12,36,14,38,14,3A,16,1A,08,18,0A,16,0A,
14,0C,16,0C,12,0E,14,0E,14,10,12,10,10,10,1
2,12,10,12,0E,12,10,14,12,14,0E,14,10,16,0E
,16,0E,18,10,18,0E,1A,6f6
220 DATA 10,1A,0E,1C,10,1E,10,20,12,22,12,2
4,14,24,14,26,16,26,18,26,1A,26,1C,26,1C,24
,1E,24,16,1E,18,1E,16,1C,18,1C,18,1A,1A,1A,
1C,1A,1C,1C,1E,1C,1E,1E,20,1E,1E,20,20,20,2
2,22,20,22,44,1A,42,1A,40,1C,3E,1E,40,1E,40
,20,3E,20,3C,22,3E,22,9b4
230 DATA 3C,24,3C,26,3C,28,3C,2A,3E,2C,40,2
C,42,2A,42,28,42,26,40,24,42,24,40,1A,44,18
,46,18,48,1A,4A,1C,4A,1E,4C,1E,4C,20,4A,20,
4C,22,4A,22,4A,24,4C,24,4A,26,4C,26,4C,28,4
A,28,4A,2A,4A,2C,4A,2E,48,30,46,30,46,2E,46
,2A,48,28,4C,2E,4E,30,100A
240 DATA 50,32,50,30,4E,2E,4E,2C,50,2C,56,1
E,54,20,52,22,54,22,52,24,50,26,50,28,52,26
,52,28,58,1E,5A,20,64,0C,66,0E,64,0E,66,10,
62,10,64,10,64,12,62,12,62,14,64,14,60,16,6
2,16,60,18,62,18,60,1A,5E,1C,60,1C,5E,1E,60
,1E,5C,20,5E,20,5C,22,11D8
250 DATA 5E,22,5C,24,5E,24,5A,26,5C,26,5A,2
8,5C,28,5A,2A,5C,2A,58,2C,5A,2C,58,2E,5A,2E
,56,30,58,30,56,32,58,32,5A,32,5C,32,5E,32,
60,32,60,30,62,30,60,22,62,22,64,24,64,26,6
6,26,64,28,66,28,64,2A,66,2A,62,2C,64,2C,62
,2E,64,2E,66,2E,68,2E,1470
```

```
260 DATA 6A,2E,6C,30,6C,32,6E,32,6C,34,6E,3
4,6C,36,6A,36,68,36,66,30,66,30,64,30,66,32
,64,32,66,34,70,26,6E,26,6C,26,6A,28,68,2A,
68,2C,6A,26,72,28,74,2A,72,2A,74,2C,72,2C,7
4,2E,72,2E,72,30,74,30,72,32,72,34,72,36,74
,38,76,38,78,36,76,36,1760
270 DATA 78,34,7A,34,78,32,7A,32,7A,30,7A,3
0,7A,2E,7C,2E,7C,2C,82,2E,80,30,80,32,7E,34
,7E,36,7C,38,7C,3A,7A,3A,76,3A,76,3A,78,3A,
84,2E,86,30,88,32,86,32,88,34,86,34,88,36,8
6,36,86,38,82,38,84,3A,86,3A,82,3C,84,3C,80
,3E,84,3E,82,40,80,40,1AE6
280 DATA 82,42,80,42,80,44,7E,44,80,46,7E,4
6,7E,48,7C,48,7E,4A,7C,4A,7C,4C,7A,4C,7C,4E
,80,4A,82,4A,82,4C,84,4C,86,58,86,5A,86,5C,
CA,5E,C8,5E,CC,5E,D2,4C,D2,4E,D4,4E,D2,50,D
4,50,D2,52,D2,54,D0,54,D2,56,D0,56,D0,58,D0
,5A,CE,5C,CE,5A,C6,5E,247C
290 DATA C6,5C,C2,52,C4,54,C2,54,C4,56,C2,5
6,C4,58,C4,5A,C2,58,C6,48,C8,46,CA,46,CC,46
,CE,48,D0,48,D2,4A,D0,4A,C6,4A,C4,4A,C2,4C,
C0,4E,C0,50,C0,52,C2,4E,C2,50,C2,52,C4,62,C
4,64,C4,66,C4,68,C2,6A,C4,6A,C2,6C,C4,6C,C2
,6E,C2,70,C4,6E,C4,70,2A42
300 DATA C2,72,C4,72,C0,74,C2,74,C0,76,C2,7
6,C0,78,C2,78,BE,7A,C0,7A,BE,7C,C0,7C,BE,7E
,C0,7E,BC,80,BE,80,BC,82,BE,82,BA,84,BC,84,
BA,86,BC,86,B8,88,BA,88,B6,8A,B8,8A,B8,8C,B
6,8C,B6,8E,B4,8E,B4,90,B6,90,B2,92,B4,92,B0
,94,B2,94,B0,96,AE,96,2F46
310 DATA AC,98,AA,98,C2,60,C0,60,BE,62,BC,6
2,BE,62,BA,64,BC,64,BA,66,B8,66,B6,68,B8,68
,B6,6A,B4,6C,B6,6C,B4,6E,B2,6E,B4,6E,B2,70,
B4,70,B0,72,B2,72,B0,74,B2,74,AE,76,B0,76,A
E,78,B0,78,AC,7A,AE,7A,AC,7C,AE,7C,AA,7E,AC
,7E,AA,80,AC,80,A8,82,2B86
320 DATA AA,82,A8,84,AA,84,A8,86,AA,86,A6,8
8,A8,88,A6,8A,A8,8A,A6,8C,A8,8C,A6,8E,A8,8E
,A6,90,A6,92,A8,94,A8,96,A6,94,C0,54,C0,56,
D2,5C,D2,5A,D4,58,D4,56,D4,54,D4,52,84,5E,8
2,5E,80,5C,82,5C,80,5A,82,5A,82,58,84,56,82
,56,88,42,8A,42,8C,32,28D0
330 DATA 8E,32,8E,34,90,34,8E,36,90,36,8C,3
8,8E,38,8C,3A,8E,3A,8A,3C,8C,3C,88,3E,8C,3E
,8A,40,8A,42,92,34,94,34,96,36,94,38,96,38,
94,3A,92,3A,94,3A,92,3C,94,3C,92,3E,94,3E,9
2,40,94,40,92,42,94,44,96,44,98,44,98,42,98
,40,9A,3E,9A,3C,9C,3A,1E76
340 DATA 9C,38,A6,3E,A4,3C,A2,3A,A2,3C,A0,3
C,A0,3E,9E,40,A0,40,9E,42,A0,42,9C,44,9E,44
,9C,46,98,44,9A,46,98,46,A8,3E,A6,40,A4,42,
A6,42,A4,44,A6,44,A4,46,A6,46,A4,48,A4,4A,A
6,4C,A8,4E,AA,4C,AC,4A,AE,48,AC,48,AE,46,AE
,44,AE,42,AE,40,AE,3E,2254
350 DATA AA,3C,AC,3C,B0,4C,B2,4E,AE,4A,A8,4
6,AA,46,6a0
360
370 RESTORE 460
380 compt=0:t=0:lig=460
390 bc=2501:fg=&6000
400 t=t+1:IF t<>26 THEN 420
410 READ a:IF a<>compt THEN 2610 ELSE GOSUB
 2630:t=1:compt=0:lig=lig+10
420 READ a:POKE fg,a:compt=compt+a
430 bc=bc-1:fg=fg+1
440 IF bc=0 THEN 1470
450 GOTO 400
460 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0
470 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,0,24
480 DATA 1,0,2,0,2,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,2,0,2,0,1,0,10
490 DATA 1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,0,1,0,22
500 DATA 1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,0,1,0,22
510 DATA 1,0,0,2,0,2,0,2,1,1,2,0,0,2,1,1,2,
0,2,0,2,0,0,1,0,22
520 DATA 1,0,1,1,1,1,1,0,1,1,0,1,1,0,1,1,0,
1,1,1,1,1,0,1,0,18
530 DATA 1,0,1,1,1,1,1,0,1,1,2,1,1,2,1,1,0,
1,1,1,1,1,0,1,0,22
540 DATA 1,0,0,2,0,0,0,0,0,0,0,1,1,0,0,0,0,
0,0,0,2,0,0,1,0,8
550 DATA 1,0,1,1,1,1,1,1,1,1,0,1,1,0,1,1,1,
1,1,1,1,1,0,1,0,20
560 DATA 1,0,1,1,1,1,1,1,1,1,0,1,1,0,1,1,1,
1,1,1,1,1,0,1,0,20
570 DATA 1,0,1,1,0,0,0,0,1,1,2,1,1,2,1,1,0,
0,0,0,1,1,0,1,0,16
580 DATA 1,0,1,1,2,1,1,2,1,1,0,1,1,0,1,1,2,
1,1,2,1,1,0,1,0,24
590 DATA 1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,
1,1,0,1,1,0,1,0,16
600 DATA 1,0,2,0,2,1,1,2,0,2,0,1,1,0,2,0,2,
1,1,2,0,2,0,1,0,24
610 DATA 1,0,1,1,1,1,1,1,1,1,0,1,1,0,1,1,1,
1,1,1,1,1,0,1,0,20
620 DATA 1,0,1,1,1,1,1,1,1,1,2,1,1,2,1,1,1,
1,1,1,1,1,0,1,0,24
630 DATA 1,0,0,2,0,2,0,2,0,0,0,1,1,0,0,0,2,
0,2,0,2,0,0,1,0,16
640 DATA 1,0,1,1,1,1,1,1,1,1,0,1,1,0,1,1,1,
1,1,1,1,1,0,1,0,20
650 DATA 1,0,1,1,1,1,1,1,1,1,0,1,1,0,1,1,1,
1,1,1,1,1,0,1,0,20
660 DATA 1,0,0,2,0,2,0,2,0,0,0,1,1,0,0,0,2,
0,2,0,2,0,0,1,0,16
670 DATA 1,0,1,1,1,1,1,1,1,1,0,1,1,0,1,1,1,
1,1,1,1,1,0,1,0,20
680 DATA 1,0,1,1,1,1,1,1,1,1,2,1,1,2,1,1,1,
1,1,1,1,1,0,1,0,24
690 DATA 1,0,0,2,0,2,0,0,0,0,0,0,0,0,0,0,0,
0,2,0,2,0,0,1,0,10
700 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,0,24
710 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0
720 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,0,24
730 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,1,0,2
740 DATA 1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,
1,1,0,1,1,0,1,0,16
750 DATA 1,0,1,1,2,1,1,2,1,1,2,1,1,2,1,1,2,
1,1,2,1,1,0,1,0,28
760 DATA 1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,
1,1,0,1,1,0,1,0,16
770 DATA 1,0,1,1,2,1,1,2,1,1,2,1,1,2,1,1,2,
1,1,2,1,1,0,1,0,28
780 DATA 1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,
1,1,0,1,1,0,1,0,16
790 DATA 1,0,1,1,2,1,1,2,1,1,2,0,0,2,1,1,2,
1,1,2,1,1,0,1,0,26
800 DATA 1,0,1,1,0,1,1,0,1,1,1,1,1,1,1,1,0,
1,1,0,1,1,0,1,0,18
810 DATA 1,0,1,1,0,1,1,0,1,1,1,1,1,1,1,1,0,
```

SERPENTINO !

```
1,1,0,1,1,0,1,0,18
820 DATA 1,0,1,1,2,0,0,2,0,0,2,0,0,2,0,0,2,
0,0,2,1,1,0,1,0,18
830 DATA 1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,0,1,0,22
840 DATA 1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,0,1,0,22
850 DATA 1,0,2,0,2,1,1,2,0,0,2,0,0,2,0,0,2,
1,1,2,0,2,0,1,0,22
860 DATA 1,0,1,1,0,1,1,0,1,1,1,1,1,1,1,1,0,
1,1,0,1,1,0,1,0,18
870 DATA 1,0,1,1,2,1,1,0,1,1,1,1,1,1,1,1,0,
1,1,2,1,1,0,1,0,22
880 DATA 1,0,1,1,0,0,2,0,0,0,2,0,0,2,0,0,0,
2,0,0,1,1,0,1,0,14
890 DATA 1,0,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,2,1,1,0,1,0,24
900 DATA 1,0,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,0,1,1,0,1,0,20
910 DATA 1,0,1,1,2,0,2,0,0,2,0,1,1,0,2,0,0,
2,0,2,1,1,0,1,0,20
920 DATA 1,0,1,1,0,1,1,1,1,1,0,1,1,0,1,1,1,
1,1,0,1,1,0,1,0,18
930 DATA 1,0,1,1,2,1,1,1,1,1,2,1,1,2,1,1,1,
1,1,2,1,1,0,1,0,26
940 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,1,0,2
950 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,0,24
960 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0
970 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,0,24
980 DATA 1,0,2,0,0,2,0,0,2,0,0,0,0,0,0,2,0,
0,2,0,0,2,0,1,0,14
990 DATA 1,0,1,1,2,1,1,2,1,1,0,1,1,0,1,1,2,
1,1,2,1,1,0,1,0,24
1000 DATA 1,0,1,1,0,1,1,0,1,1,2,1,1,2,1,1,0
,1,1,0,1,1,0,1,0,20
1010 DATA 1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,2,0,1,0,6
1020 DATA 1,0,1,1,2,1,1,0,1,1,0,1,1,0,1,1,0
,1,1,2,1,1,0,1,0,20
1030 DATA 1,0,1,1,0,1,1,0,1,1,2,1,1,2,1,1,0
,1,1,0,1,1,0,1,0,20
1040 DATA 1,0,2,0,0,2,0,0,0,2,0,0,0,0,2,0,0
,0,2,0,0,2,0,1,0,14
1050 DATA 1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0
,1,1,0,1,1,0,1,0,16
1060 DATA 1,0,1,1,0,1,1,2,1,1,0,1,1,0,1,1,2
,1,1,0,1,1,0,1,0,20
1070 DATA 1,0,2,0,0,0,0,0,0,0,2,0,0,2,0,0,0
,0,0,0,0,2,0,1,0,10
1080 DATA 1,0,1,1,2,1,1,0,1,1,0,1,1,0,1,1,0
,1,1,2,1,1,0,1,0,20
1090 DATA 1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0
,1,1,0,1,1,0,1,0,16
1100 DATA 1,0,0,0,0,0,2,0,2,0,0,0,0,0,0,2,0
,2,0,0,0,0,0,1,0,10
1110 DATA 1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0
,1,1,0,1,1,0,1,0,16
1120 DATA 1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0
,1,1,0,1,1,0,1,0,16
1130 DATA 1,0,2,0,0,0,0,0,0,0,2,0,0,2,0,0,0
,0,0,0,0,2,0,1,0,10
1140 DATA 1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0
,1,1,0,1,1,0,1,0,16
1150 DATA 1,0,1,1,0,1,1,2,1,1,0,1,1,0,1,1,2
,1,1,0,1,1,0,1,0,20
1160 DATA 1,0,2,0,0,0,0,0,0,0,2,0,0,2,0,0,0
,0,0,0,0,2,0,1,0,10
1170 DATA 1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0
,1,1,0,1,1,0,1,0,16
1180 DATA 1,0,1,1,0,1,1,2,1,1,0,1,1,0,1,1,2
,1,1,0,1,1,0,1,0,20
1190 DATA 1,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,2,0,0,0,1,0,6
1200 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,0,24
1210 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0
1220 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,0,24
1230 DATA 1,2,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,2,0,0,2,1,0,10
1240 DATA 1,0,1,1,1,1,1,1,1,1,0,1,1,0,1,1,1
,1,1,1,1,1,0,1,0,20
1250 DATA 1,0,1,1,1,1,1,1,1,1,0,1,1,0,1,1,1
,1,1,1,1,1,0,1,0,20
1260 DATA 1,2,1,1,2,0,2,0,2,0,0,1,1,0,0,2,0
,2,0,2,1,1,2,1,0,24
1270 DATA 1,0,1,1,0,1,1,1,1,1,0,1,1,0,1,1,1
,1,1,0,1,1,0,1,0,18
1280 DATA 1,0,1,1,2,1,1,1,1,1,0,1,1,0,1,1,1
,1,1,2,1,1,0,1,0,22
1290 DATA 1,2,1,1,0,1,1,2,0,2,0,1,1,0,2,0,2
,1,1,0,1,1,2,1,0,24
1300 DATA 1,0,1,1,2,1,1,0,1,1,0,1,1,0,1,1,0
,1,1,2,1,1,0,1,0,20
1310 DATA 1,0,1,1,0,1,1,2,1,1,0,1,1,0,1,1,2
,1,1,0,1,1,0,1,0,20
1320 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,1,0,2
1330 DATA 1,1,1,1,1,1,1,1,1,1,0,1,1,0,1,1,1
,1,1,1,1,1,1,1,0,22
1340 DATA 1,1,1,1,1,1,1,1,1,1,2,1,1,2,1,1,1
,1,1,1,1,1,1,1,0,26
1350 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,1,0,2
1360 DATA 1,0,1,1,0,1,1,2,1,1,0,1,1,0,1,1,2
,1,1,0,1,1,0,1,0,20
1370 DATA 1,0,1,1,2,1,1,0,1,1,0,1,1,0,1,1,0
,1,1,2,1,1,0,1,0,20
1380 DATA 1,2,1,1,0,1,1,2,0,2,0,1,1,0,2,0,2
,1,1,0,1,1,2,1,0,24
1390 DATA 1,0,1,1,2,1,1,1,1,1,0,1,1,0,1,1,1
,1,1,2,1,1,0,1,0,22
1400 DATA 1,0,1,1,0,1,1,1,1,1,0,1,1,0,1,1,1
,1,1,0,1,1,0,1,0,18
1410 DATA 1,2,1,1,2,0,2,0,2,0,0,1,1,0,0,2,0
,2,0,2,1,1,2,1,0,24
1420 DATA 1,0,1,1,1,1,1,1,1,1,0,1,1,0,1,1,1
,1,1,1,1,1,0,1,0,20
1430 DATA 1,0,1,1,1,1,1,1,1,1,0,1,1,0,1,1,1
,1,1,1,1,1,0,1,0,20
1440 DATA 1,2,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,2,0,0,2,1,0,10
1450 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,0,24
1460 '
1470 PRINT"OK"
1480 '
1490 '
1500 RESTORE 1560:compt=0:lig=1560
1510 FOR i=&7A00 TO &7A5A
1520 READ a:IF a<256 THEN 1540
1530 IF compt<>a THEN GOTO 2610 ELSE GOSUB
2630:compt=0:lig=lig+10:READ a
```

```
1540 compt=compt+a:POKE i,a
1550 NEXT
1560 DATA 83,67,79,82,69,58,72,73,83,67,32,
58,66,79,78,85,83,58,80,65,84,82,73,67,69,6
9,82,73,69,65,85,83,69,82,58,84,65,66,58,28
00
1570 DATA 71,65,77,69,32,32,45,32,32,79,86,
69,82,65,80,80,85,89,69,82,32,32,32,32,32,3
2,83,85,82,1763
1580 DATA 32,32,32,32,32,32,32,85,78,69,32,
32,32,32,32,32,84,79,85,67,72,69,1104
1590 PRINT"OK"
1600 RESTORE 1780
1610 compt=0:lig=1780
1620 FOR i=&7A5A TO &7C2C STEP 9
1630 READ a:IF a>=0 THEN 1660
1640 a=ABS(a):IF a<>compt THEN GOTO 2610
1650 GOSUB 2630:compt=0:lig=lig+10:READ a
1660 POKE i,1
1670 POKE i+1,0:POKE i+2,0
1680 d$=HEX$(a,4):l$="&"+LEFT$(d$,2):r$="&"
+RIGHT$(d$,2):compt=compt+a
1690 POKE i+3,VAL(r$)
1700 POKE i+4,VAL(l$)
1710 POKE i+5,0
1720 READ a:POKE i+6,VAL("&"+HEX$(a)):compt
=compt+a
1730 READ a:d$=HEX$(a,4):l$="&"+LEFT$(d$,2)
:r$="&"+RIGHT$(d$,2):compt=compt+a
1740 POKE i+7,VAL(r$)
1750 POKE i+8,VAL(l$)
1760 NEXT:READ a:IF compt<>ABS(a) THEN 2610
 ELSE GOSUB 2630
1770 PRINT"OK"
1780 DATA 200,15,8,100,15,3,50,15,5,-411
1790 DATA 119,15,20,106,15,20,89,15,20,119,
15,10,106,15,10,89,15,40,60,15,20,53,15,20,
45,15,20,45,15,10,53,15,10,60,15,30,119,10,
10,119,9,18,-1639
1800 DATA 119,8,22,119,7,25,119,5,27,119,2,
28,-600
1810 DATA 1000,15,5,1000,13,4,1000,10,3,100
0,5,2,1000,2,1,-5060
1820 DATA 758,15,100,0,0,5,758,15,50,0,0,5,
758,15,25,0,0,1,-2505
1830 DATA 758,15,110,0,0,2,638,15,60,676,15
,25,0,0,5,758,15,55,0,0,3,758,15,25,0,0,3,7
58,15,60,0,0,3,851,15,30,0,0,1,-5684
1840 DATA 758,15,60,758,13,50,758,10,40,758
,7,30,758,4,25,758,2,20,758,1,10,-5593
1850 '
1860 '
1870 '
1880 RESTORE 1960:compt=0:lig=1960
1890 FOR i=&9000 TO &931D
1900 READ a$
1910 a=VAL("&"+a$):IF a<256 THEN 1940
1920 IF compt<>a THEN GOTO 2610
1930 i=i-1:GOSUB 2630:compt=0:lig=lig+10:GO
TO 1950
1940 POKE i,a:compt=compt+a
1950 NEXT:PRINT:PRINT"OK":PRINT
1960 DATA DD,7E,00,06,09,B8,C8,DD,34,00,3E,
03,CD,90,BB,3E,00,CD,96,BB,21,16,20,CD,75,B
B,3E,F3,DD,46,00,CD,5A,BB,05,C8,18,F9,DD,34
,0A,DD,7E,0A,06,0A,B8,20,43,DD,36,0A,00,DD,
34,0B,DD,7E,0B,06,0A,B8,20,34,DD,36,0B,00,D
D,34,0C,DD,7E,0C,06,0A,B8,1DE8
1970 DATA 20,25,DD,36,0C,00,DD,34,0D,DD,7E,
0D,06,0A,B8,20,16,DD,36,0D,00,DD,34,0E,DD,7
E,0E,06,0A,B8,20,07,DD,36,0E,00,DD,34,0F,3E
,03,CD,90,BB,3E,00,CD,96,BB,21,02,22,CD,75,
BB,06,06,11,0F,7F,1A,C6,30,CD,5A,BB,1B,05,2
0,F6,DD,7E,0F,DD,46,06,B8,1AD5
1980 DATA 38,47,20,36,DD,7E,0E,DD,46,05,B8,
38,3C,20,2B,DD,7E,0D,DD,46,04,B8,38,31,20,2
0,DD,7E,0C,DD,46,03,B8,38,26,20,15,DD,7E,0B
,DD,46,02,B8,38,1B,20,0A,DD,7E,0A,DD,46,01,
B8,38,10,C8,21,01,7F,11,0A,7F,06,06,1A,77,2
3,13,05,20,F9,3E,03,CD,90,1980
1990 DATA BB,3E,00,CD,96,BB,21,04,22,CD,75,
BB,06,06,11,06,7F,1A,C6,30,CD,5A,BB,1B,05,2
0,F6,C9,FE,FF,20,06,CD,29,91,3E,01,C9,DD,46
,32,B8,C2,1B,92,CD,26,90,CD,D8,92,DD,36,13,
00,CD,3D,91,DD,35,25,28,6E,3E,02,C9,DD,34,1
3,DD,7E,13,FE,15,20,04,DD,2113
2000 DATA 36,13,14,DD,7E,13,FE,14,C0,DD,35,
16,DD,7E,16,FE,00,20,1C,DD,36,16,0A,DD,35,1
5,DD,7E,15,FE,00,20,0E,DD,36,15,0A,DD,35,14
,DD,7E,14,FE,00,28,2E,3E,03,CD,90,BB,3E,00,
CD,96,BB,21,07,22,CD,75,BB,DD,7E,14,C6,2F,C
D,5A,BB,DD,7E,15,C6,2F,CD,1FF5
2010 DATA 5A,BB,DD,7E,16,C6,2F,CD,5A,BB,3E,
02,CD,90,BB,C9,C3,1B,92,D1,DD,7E,17,FE,67,2
0,0D,DD,36,17,60,DD,36,18,00,CD,00,90,18,11
,DD,66,17,DD,6E,18,01,70,02,ED,4A,DD,74,17,
DD,75,18,DD,7E,27,FE,01,28,05,DE,14,DD,77,2
7,DD,7E,32,FE,FE,20,04,DD,22DC
2020 DATA 36,32,F8,DD,34,32,CD,95,92,DD,36,
33,02,DD,35,16,28,0B,CD,63,91,CD,26,90,CD,F
8,92,18,F0,DD,34,16,DD,35,15,28,0B,CD,63,91
,CD,35,90,CD,F8,92,18,F0,DD,34,15,DD,35,14,
28,0B,CD,63,91,CD,44,90,CD,F8,92,18,F0,DD,3
6,25,32,CD,FD,92,C3,38,80,2591
2030 DATA D1,CD,BD,83,CD,06,93,DD,46,1F,DD,
7E,1E,B8,20,F1,2A,F2,7F,CD,75,BB,3E,20,CD,5
A,BB,DD,7E,00,FE,01,28,0E,DD,35,00,CD,0A,90
,3E,20,CD,5A,BB,C3,95,80,CD,6C,BB,3E,02,CD,
90,BB,21,0D,0D,CD,75,BB,11,27,7A,06,0D,1A,C
D,5A,BB,13,05,20,F8,CD,14,2317
2040 DATA 93,3E,03,CD,90,BB,21,24,02,CD,75,
BB,11,34,7A,06,26,1A,CD,5A,BB,13,05,20,F8,C
D,03,BB,CD,06,BB,CD,6C,BB,DD,36,25,32,DD,36
,27,F1,C3,10,80,DD,34,09,DD,7E,09,FE,0A,20,
15,DD,36,09,00,DD,34,08,DD,7E,08,FE,0A,20,0
7,DD,36,08,00,DD,34,07,3E,1E2F
2050 DATA 03,CD,90,BB,21,18,20,CD,75,BB,DD,
7E,07,C6,30,CD,5A,BB,DD,7E,08,C6,30,CD,5A,B
B,DD,7E,09,C6,30,CD,5A,BB,C9,21,5A,7A,16,03
,E5,D5,CD,AA,BC,30,FB,3E,00,CD,B3,BC,D1,E1,
01,09,00,ED,4A,15,20,EA,DD,21,00,7F,C9,C9,C
D,5A,BB,C9,21,75,7A,16,12,258F
2060 DATA CD,DD,92,C9,21,17,7B,16,05,CD,DD,
92,06,FF,05,20,FD,C9,21,44,7B,16,1A,CD,DD,9
2,C9,d14
2070 '
2080 '
2090 RESTORE 2170
2100 compt=0:lig=2170
2110 FOR i=&9400  TO &942F
2120 READ a$
2130 a=VAL("&"+a$):IF a<256 THEN 2150
2140 IF a<>compt THEN GOTO 2610 ELSE GOSUB
2630:GOTO 2160
2150 POKE i,a:compt=compt+a
2160 NEXT:PRINT:PRINT"OK":PRINT
```

SERPENTINO !

```
2170 DATA 3E,03,CD,DE,BB,01,00,5A,0A,26,00,
6F,C5,01,90,01,ED,4A,EB,C1,03,0A,26,00,6F,C
5,01,5A,00,ED,4A,CD,EA,BB,C1,03,3E,18,B9,20
,DF,3E,5F,B8,20,DA,C9,1330
2180 '
2190 '
2200 RESTORE 2260:compt=0:lig=2260:FOR i=&8
000 TO &844D
2210 READ a$:a=VAL("&"+a$):IF a<256 THEN 22
40
2220 IF a<>compt THEN GOTO 2610
2230 GOSUB 2630:compt=0:lig=lig+10:GOTO 221
0
2240 POKE i,a:compt=compt+a
2250 NEXT:READ a$:a=VAL("&"+a$):IF a<>compt
 THEN 2610 ELSE GOSUB 2630:PRINT:PRINT"OK"
2260 DATA DD,21,00,7F,21,00,7F,3E,00,06,08,
77,23,05,20,FB,21,0A,7F,3E,00,06,07,77,23,0
5,20,FB,DD,36,32,F9,DD,36,00,03,DD,36,07,00
,DD,36,08,00,DD,36,09,01,DD,36,17,60,DD,36,
18,00,DD,56,17,DD,5E,18,21,01,00,01,71,02,C
5,24,1A,06,01,B8,20,0D,3E,158D
2270 DATA FF,DD,77,28,E5,3E,01,CD,90,BB,18,
19,06,00,B8,20,08,3E,20,DD,77,28,E5,18,0C,D
D,7E,32,DD,77,28,E5,3E,03,CD,90,BB,E1,E5,D5
,CD,75,BB,DD,7E,28,CD,5A,BB,D1,E1,C1,3E,19,
BC,20,03,26,00,2C,13,0B,3E,00,B8,20,B4,3E,0
0,B9,20,AF,DD,36,14,09,DD,21AB
2280 DATA 36,15,09,DD,36,16,09,21,18,0F,22,
F0,7F,21,18,0B,22,F2,7F,3E,02,CD,90,BB,2A,F
2,7F,CD,75,BB,3E,F8,CD,5A,BB,3E,F1,CD,5A,BB
,3E,F1,CD,5A,BB,3E,F1,CD,5A,BB,3E,F7,CD,5A,
BB,3E,00,CD,90,BB,3E,01,CD,96,BB,21,02,1B,C
D,75,BB,06,06,11,00,7A,CD,22DA
2290 DATA 23,81,21,04,1B,CD,75,BB,06,06,11,
06,7A,CD,23,81,21,07,1B,CD,75,BB,06,06,11,0
C,7A,CD,23,81,21,16,1B,CD,75,BB,06,04,11,1F
,7A,CD,23,81,21,18,1B,CD,75,BB,06,04,11,23,
7A,CD,23,81,18,09,1A,CD,5A,BB,13,05,20,F8,C
9,3E,01,CD,96,BB,3E,02,CD,19B1
2300 DATA 90,BB,21,0A,1B,CD,75,BB,06,07,11,
12,7A,CD,23,81,21,14,22,CD,75,BB,06,06,11,1
9,7A,CD,23,81,CD,0A,90,CD,E3,90,CD,74,90,DD
,36,14,0A,DD,36,15,0A,DD,36,16,0A,CD,63,91,
CD,00,94,CD,B4,92,DD,36,1F,04,DD,36,1E,00,D
D,36,20,03,DD,36,22,00,DD,1E10
2310 DATA 36,46,00,21,00,7E,3E,03,06,06,77,
23,05,20,FB,3E,02,CD,90,BB,DD,36,21,01,CD,2
4,BB,CB,47,20,24,CB,4F,20,38,CB,57,20,4C,CB
,5F,20,61,DD,36,21,00,DD,7E,20,FE,00,28,6F,
FE,01,CA,82,82,FE,02,CA,DE,82,C3,34,83,2A,F
0,7F,2D,CD,75,BB,CD,60,BB,1FBC
2320 DATA FE,FF,28,DE,DD,7E,20,FE,01,CA,82,
82,18,4A,2A,F0,7F,2C,CD,75,BB,CD,60,BB,FE,F
F,28,C6,DD,7E,20,FE,00,28,35,C3,82,82,2A,F0
,7F,25,CD,75,BB,CD,60,BB,FE,FF,28,AE,DD,7E,
20,FE,03,CA,34,83,C3,DE,82,2A,F0,7F,24,CD,7
5,BB,CD,60,BB,FE,FF,28,95,2BD1
2330 DATA DD,7E,20,FE,02,CA,DE,82,C3,34,83,
DD,36,20,00,2A,F0,7F,2D,CD,75,BB,CD,60,BB,F
E,20,28,12,CD,03,91,FE,02,28,0E,DD,7E,21,FE
,01,CA,95,81,C3,8C,83,CD,BD,83,2A,F0,7F,CD,
75,BB,3E,F1,CD,5A,BB,2A,F0,7F,2D,22,F0,7F,C
D,75,BB,3E,F0,CD,5A,BB,CD,28FE
2340 DATA 37,91,26,7E,DD,6E,1F,23,3E,FF,BD,
20,02,2E,00,36,00,DD,75,1F,CD,3D,91,C3,3F,8
4,DD,36,20,01,2A,F0,7F,2C,CD,75,BB,CD,60,BB
,FE,20,28,12,CD,03,91,FE,02,28,0E,DD,7E,21,
FE,01,CA,95,81,C3,8C,83,CD,BD,83,2A,F0,7F,C
D,75,BB,3E,F1,CD,5A,BB,2A,23E3
2350 DATA F0,7F,2C,22,F0,7F,CD,75,BB,3E,F3,
CD,5A,BB,CD,37,91,26,7E,DD,6E,1F,23,3E,FF,B
D,20,02,2E,00,36,01,DD,75,1F,CD,3D,91,C3,3F
,84,DD,36,20,02,2A,F0,7F,25,CD,75,BB,CD,60,
BB,FE,20,28,12,CD,03,91,FE,02,28,0E,DD,7E,2
1,FE,01,CA,95,81,C3,8C,83,2409
2360 DATA CD,BD,83,2A,F0,7F,25,22,F0,7F,CD,
75,BB,3E,F5,CD,5A,BB,3E,F1,CD,5A,BB,CD,37,9
1,26,7E,DD,6E,1F,23,3E,FF,BD,20,02,2E,00,36
,02,DD,75,1F,CD,3D,91,C3,3F,84,DD,36,20,03,
2A,F0,7F,24,CD,75,BB,CD,60,BB,FE,20,28,11,C
D,03,91,FE,02,28,0D,DD,7E,2479
2370 DATA 21,FE,01,CA,95,81,18,35,CD,BD,83,
2A,F0,7F,CD,75,BB,3E,F1,CD,5A,BB,3E,F7,CD,5
A,BB,2A,F0,7F,24,22,F0,7F,CD,37,91,26,7E,DD
,6E,1F,23,3E,FF,BD,20,02,2E,00,36,03,DD,75,
1F,CD,3D,91,C3,3F,84,26,7D,DD,6E,22,23,3E,F
F,BD,20,02,2E,00,DD,75,22,2332
2380 DATA 7E,4F,DD,7E,20,CB,4F,28,03,79,18,
03,79,C6,02,FE,00,CA,C4,81,FE,01,CA,DC,81,F
E,02,CA,F4,81,C3,0D,82,26,7E,DD,6E,1E,23,3E
,FF,BD,20,02,2E,00,DD,75,1E,7E,FE,00,28,0A,
FE,01,28,21,FE,02,28,38,18,4B,2A,F2,7F,CD,7
5,BB,3E,20,CD,5A,BB,2A,F2,21BD
2390 DATA 7F,2D,22,F2,7F,CD,75,BB,3E,F2,CD,
5A,BB,C9,2A,F2,7F,CD,75,BB,3E,20,CD,5A,BB,2
A,F2,7F,2C,22,F2,7F,CD,75,BB,3E,F4,CD,5A,BB
,C9,2A,F2,7F,25,22,F2,7F,CD,75,BB,3E,F6,CD,
5A,BB,3E,20,CD,5A,BB,C9,2A,F2,7F,CD,75,BB,3
E,20,CD,5A,BB,3E,F8,CD,5A,2AC7
2400 DATA BB,2A,F2,7F,24,22,F2,7F,C9,DD,46,
27,DD,4E,27,0D,20,FD,05,20,F7,C3,95,81,B91
2410 MODE 2
2420 PRINT:PRINT:PRINT
2430 PRINT"- Si vous utilisez une disquette
: "
2440 PRINT"-------------------------------------
--------"
2450 PRINT"     *  Mettre la disquette cont
enant le programme: - SERPENT.BAS -"
2460 PRINT:PRINT:PRINT:PRINT:PRINT
2470 PRINT"- Si vous utilisez une cassette:
"
2480 PRINT"-------------------------------------
--------"
2490 PRINT"     *  Mettre les fichiers cree
s par ce programme a la suite du programme:
          - SERPENT.BAS -"
2500 PRINT:PRINT"     *  Puis appuyez sur
REC & PLAY-"
2510 PRINT:PRINT:PRINT:PRINT:PRINT
2520 PRINT"  Lorsque vous etes pret APPUYEZ
 SUR UNE TOUCHE"
2530 CALL &BB06:MEMORY &A000
2540 SAVE"!dessin",b,&5A00,&520
2550 SAVE"!table",b,&6000,2501
2560 SAVE"!musiq",b,&7A00,&200
2570 SAVE"!serp1",b,&9000,&330
2580 SAVE"!serp2",b,&9400,&30
2590 SAVE"!serp3",b,&8000,&600
2600 END
2610 PRINT CHR$(7):PRINT"ERREUR ....... lig
ne:";lig
2620 END
2630 PRINT"ligne :";lig;"      OK"
2640 RETURN
```



VÉRIFICATEUR V3

POUR PCW 8256/8512

Ce vérificateur basic ne comporte malheureusement pas tous les avantages de son homologue sur CPC. Cependant, en attendant mieux, il nous paraissait indispensable de vous donner dès le départ une bonne base de travail. Rassurez-vous, nous tâcherons très prochainement de vous fournir une véritable copie de l'original, avec tout ce que cela suppose...

Pour utiliser Vérif V3, rien de plus simple. Tapez le listing en prenant bien garde de ne pas commettre d'erreurs de saisie (il est recommandé de s'y prendre à deux fois avant de s'avouer vaincu par un syntaxe error...), et sauvegardez-le sur une disquette sûre. Vous n'aurez plus, dès à présent qu'à taper vos listings, tout en étant décontractés, et pour cause, et, TRÈS IMPORTANT, à les sauvegarder en ASCII. Pour ce faire, procédez de la façon suivante :
tapez SAVE "nomprog$. bas", a

— nomprog$ désignant le nom du programme, et le (a), signifiant que le mode de sauvegarde doit être impérativement ASCII. Cette opération réalisée, lancez le vérificateur par RUN, suivi de l'appellation sous laquelle vous l'aurez sauvegardé. Aussitôt, le programme vous demandera de lui fournir le nom du programme à vérifier ; exemple : nomprog$,bas (n'oubliez pas le "bas", sans lequel le vérificateur ne pourrait trouver sa source), indiquez-lui ensuite si vous désirez obtenir un listing écran ou imprimante, et le tour est joué.

```
10 cls$=CHR$(27)+"E"+CHR$(27)+"H"
20 ceu$=CHR$(27)+"R"+CHR$(0)
30 cf$=CHR$(27)+"R"+CHR$(1)
40 DEF FN locate$(x,y)=CHR$(27)+"Y"
+CHR$(32+y)+CHR$(32+x)
50 PRINT cls$
60 x=28:y=3:GOSUB 350:PRINT "VERIFI
CATEUR V3 pour PCW 8256-8512"
70 x=36:y=5:GOSUB 350:PRINT CHR$(16
4);" AMSTRAD MAGAZINE"
80 x=26:y=6:GOSUB 350:PRINT "auteur
s: Eric MISTELET & Georges BRIZE"
90 WIDTH LPRINT 80
100 x=1:y=9: GOSUB 350:INPUT "Nom d
u programme à vérifier :";prgm$
110 x=1:y=10: GOSUB 350:INPUT "Sort
ie sur (E)cran ou (I)mprimante ";r$
120 r$=UPPER$(r$)
130 IF r$="I" THEN prt=1:GOTO 160
140 IF r$<>"E" THEN 110
150 PRINT cls$
160 OPEN "i",1,prgm$
170 WHILE EOF (1)<>-1
180 LINE INPUT #1,a$
190 n=0:v=0
200 FOR I=1 TO LEN(a$)
210 n=n+ASC(UPPER$(MID$(a$,i,1)))
220 NEXT
230 FOR i=1 TO LEN(a$)
240 c$=MID$(a$,i,1):IF prt=1 THEN L
PRINT c$;:ELSE PRINT c$;
250 d=i/35: IF d=INT(d) THEN IF d=1
 THEN GOSUB 300:ELSE GOSUB 330
260 NEXT
270 IF v=0 THEN GOSUB 300:ELSE IF i
>36 THEN GOSUB 330
280 WEND
290 END
300 IF prt THEN LPRINT TAB(37);ceu$
+CHR$(91);cf$;n;ceu$+CHR$(93);:LPRI
NT cf$:ELSE PRINT TAB(37);CHR$(162)
;n;CHR$(166)
310 v=1
320 RETURN
330 IF prt THEN LPRINT:ELSE PRINT
340 RETURN
350 PRINT FN locate$(x,y);:RETURN
```



OTHELLO *(PCW)*

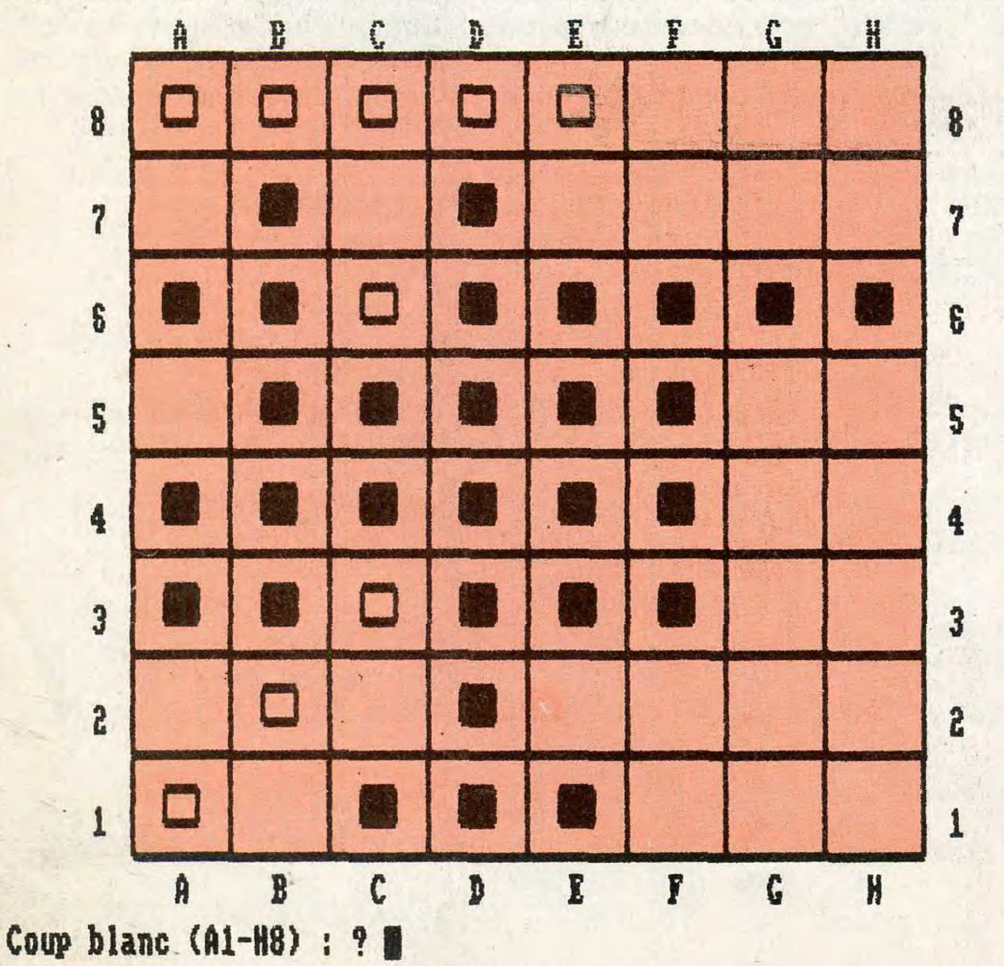
Pour ceux qui l'ignorent, le jeu d'Othello est une joute intellectuelle entre deux adversaires, se déroulant sur une grille de huit cases sur huit (coordonnées A1 à H8). Débutant la partie avec deux pions de la couleur de son choix (noir ou blanc), chaque joueur doit s'efforcer de prendre un maximum de positions concurrentes par un entourage judicieux. Le gagnant est celui qui, en fin de partie, totalise le plus de points. Ce programme sympathique, résultat d'un travail personnel (non recopie ou adaptation d'un listing déjà publié), permet de jouer à deux (affichage, validité des coups et comptage des points) ou seul contre l'ordinateur. Dans ce dernier cas (pions blancs pour le joueur), deux types de jeu sont proposés :

— Avec algorithme « standard » : la force de jeu est constante.

— Avec algorithme « évolutif » : le programme acquiert une expérience en fonction des parties précédentes. Un fichier (EVOL) alors créé sur disquette, permet la mémorisation des coups du vainqueur.

L'algorithme utilisé pour le choix d'un coup parmi ceux possibles, est basé sur deux facteurs importants du jeu d'Othello : la position (stratégie) et le nombre de points marqués. Pour guider le choix de position, un tableau (8,8) de nom cherche%(i,j) est initialisé avec une pondération pour chaque case figée dans le cas de l'algorithme standard, et sans cesse modifiée à l'aide du fichier disque dans le cas de l'algorithme évolutif. Le nombre de points est, pour chaque case jouable, calculé avant de jouer. Le coup final retenu est celui qui, à pondération maximum, donne le gain maximum. En fin de partie, seul le nombre de points marqués détermine le coup joué. A titre d'exemple de pondération, les quatre coins A1, A8, H1 et H8, qui une fois pris sont acquis jusqu'à la fin du jeu, ont (dans le cas de l'algorithme standard) la pondération maximum de 8. Les coups B2, B7, G2 et G7 « ouvrant » ces coins, ont la pondération maximum de 3.

Aurélia De Geeter



coup n° 18 scores

blancs : 29
noirs : 9

coups permis - points

B1 2 C2 2 C7 2

- coups joués -

D3-B3-F3-F6-C5-B7-F5-
C1-D7-B6-A6-A4-H6-A3-
B8-E1-D8-

C3-E3-F4-C4-C6-D6-D2-
A8-E6-B5-B4-G6-B2-E8-
D1-A1-C8-

```
10 '                                    [ 168 ]
20 REM *** programme othello le 15/ [ 3819 ]
06/87 Aurélia De Geeter ***
30 '                                    [ 170 ]
40 MEMORY &HEFFF                        [ 1026 ]
50 REM * routine de définition de c [ 2998 ]
aractère *
60 DIM cherche%(8,8)                    [ 1156 ]
70 DIM ecran%(10,10):DIM ecrposs%(1 [ 2309 ]
0,10)
80 DIM cqupbc%(80):DIM coupnr%(80)  [ 2053 ]
90 DATA &H21,0,&HB8,&HED,&H4B,8,&HF [ 11217 ]
0,&HCB,&H20,&HCB,&H21,&H38,&H0E,&HC
B,&H20,&HCB,&H21,&H38,&H0B,&HCB,&H2
0,&HCB,&H21,&H38,8,&H18,7,4,&H18,&H
EF,4,&H18,&HF2,4,9,6,8,&H11,0,&HF0,
&H1A,&H77,&H13,&H23,5,&H20,&HF9,&HC
9
100 DATA &H01,&H0A,&HF0,&HCD,&H5A,& [ 2897 ]
HFC,&HE9,&H00,&HC9
110 FOR i=&HF00A TO &HF00A+47:READ  [ 3051 ]
n:POKE i,n:NEXT
120 FOR i=&HF040 TO &HF040+8:READ n [ 2975 ]
:POKE i,n:NEXT
130 defcar%=&HF040                      [ 1027 ]
140 REM * déf du coin haut gauche p [ 2571 ]
lein *
150 DATA 0,0,0,&H0F,&H1F,&H1F,&H1F, [ 2457 ]
&H1F,&HA6,0
160 FOR i=&HF000 TO &HF000+9:READ n [ 2972 ]
:POKE i,n:NEXT
170 CALL defcar%                        [ 958 ]
180 REM * déf du coin haut droite p [ 2601 ]
lein *
190 DATA 0,0,0,&HF0,&HF8,&HF8,&HF8, [ 2502 ]
&HF8,&HAC,0
200 FOR i=&HF000 TO &HF000+9:READ n [ 2967 ]
:POKE i,n:NEXT
210 CALL defcar%                        [ 953 ]
220 REM * déf du coin bas droite pl [ 2472 ]
ein*
230 DATA &HF8,&HF8,&HF8,&HF8,&HF0,0 [ 2487 ]
,0,0,&HA9,0
240 FOR i=&HF000 TO &HF000+9:READ n [ 2971 ]
:POKE i,n:NEXT
250 CALL defcar%                        [ 957 ]
260 REM * déf du coin bas gauche pl [ 2482 ]
ein *
270 DATA &H1F,&H1F,&H1F,&H1F,&H0F,0 [ 2457 ]
,0,0,&HA3,0
280 FOR i=&HF000 TO &HF000+9:READ n [ 2975 ]
:POKE i,n:NEXT
290 CALL defcar%                        [ 961 ]
300 REM * déf du pavé bas demi-plei [ 2405 ]
n *
310 DATA 0,0,0,&HFF,&HFF,&HFF,&HFF, [ 2572 ]
&HFF,&HAA,0
320 FOR i=&HF000 TO &HF000+9:READ n [ 2970 ]
:POKE i,n:NEXT
330 CALL defcar%                        [ 956 ]
340 REM * déf du pavé haut demi-ple [ 2501 ]
in *
350 DATA &HFF,&HFF,&HFF,&HFF,&HFF,0 [ 2577 ]
,0,0,&HAB,0
360 FOR i=&HF000 TO &HF000+9:READ n [ 2974 ]
:POKE i,n:NEXT
370 CALL defcar%                        [ 960 ]
380 REM * init *                        [ 903 ]
390 esc$=CHR$(27):cls$=esc$+"0"+esc [ 4222 ]
$+"E"+esc$+"H":iv$=esc$+"p":vi$=esc
$+"q"
400 lig$="   A     B     C     D    [ 2329 ]
 E     F     G     H"
410 WIDTH 255:scoreb%=2:scoren%=2:c [ 2585 ]
oup%=1
```

```
420 DEF FNpos$(x,y)=esc$+"Y"+CHR$(3 [ 2892 ]
2+x)+CHR$(32+y)
430 PRINT cls$;                         [ 933 ]
440 xfd%=35:yfd%=0:yfc%=2:xfe%=48:y [ 3074 ]
fe%=2:GOSUB 1520
450 PRINT FNpos$(1,36);iv$;" - Othe [ 2766 ]
llo - ";vi$;
460 PRINT FNpos$(10,5);"- 0 - Fin"; [ 1957 ]
470 PRINT FNpos$(12,5);"- 1 - Deux  [ 2639 ]
joueurs";
480 PRINT FNpos$(14,5);"- 2 - Jeu s [ 2597 ]
tandard";
490 PRINT FNpos$(16,5);"- 3 - Jeu é [ 2684 ]
volutif";
500 PRINT FNpos$(28,0);"Choix (0 à  [ 2342 ]
3) : ";
510 a$=INKEY$:WHILE (a$<"0") OR (a$ [ 3177 ]
>"3"):a$=INKEY$:WEND
520 IF a$="0" THEN PRINT esc$+"e";: [ 2199 ]
END
530 indjeu%=ASC(a$)-48:IF indjeu%=1 [ 2628 ]
 THEN 560
540 IF indjeu%=2 THEN FOR i=0 TO 7: [ 4770 ]
FOR j=0 TO 7:READ cherche%(i,j):NEX
T:NEXT
550 IF indjeu%=3 THEN IF FIND$("evo [ 11697 ]
l")="" THEN FOR i=0 TO 7:FOR j=0 TO
 7:cherche%(i,j)=1:NEXT:NEXT:ELSE O
PEN "I",1,"evol":FOR i=0 TO 7:FOR j
=0 TO 7:INPUT # 1,cherche%(i,j):NEX
T:NEXT:CLOSE 1
560 GOSUB 1050                          [ 801 ]
570 bc%=1:x%=ASC("5")-48:y%=ASC("D" [ 2786 ]
)-64:GOSUB 1180
580 icoupbc%=0:icoupnr%=0               [ 1600 ]
590 DATA 8,3,7,7,7,7,3,8                [ 1246 ]
600 DATA 3,3,4,4,4,4,3,3                [ 1216 ]
610 DATA 7,4,7,7,7,7,4,7                [ 1239 ]
620 DATA 7,4,7,7,7,7,4,7                [ 1240 ]
630 DATA 7,4,7,7,7,7,4,7                [ 1241 ]
640 DATA 7,4,7,7,7,7,4,7                [ 1242 ]
650 DATA 3,3,4,4,4,4,3,3                [ 1221 ]
660 DATA 8,3,7,7,7,7,3,8                [ 1244 ]
670 x%=ASC("4")-48:y%=ASC("E")-64:G [ 2449 ]
OSUB 1180
680 bc%=2:x%=ASC("4")-48:y%=ASC("D" [ 2788 ]
)-64:GOSUB 1180
690 x%=ASC("5")-48:y%=ASC("E")-64:G [ 2452 ]
OSUB 1180
700 xfd%=64:yfd%=0:yfc%=2:xfe%=88:y [ 3082 ]
fe%=5:GOSUB 1520
710 nbcoup%=0:PRINT FNpos$(1,65);iv [ 8950 ]
$;STRING$(24," ");FNpos$(1,67);"cou
p n°";nbcoup%;"  scores";vi$;FNpos$
(3,66);"blancs : ";FNpos$(4,66);"no
irs  : ";
720 REM * Cadre des coups possibles [ 2346 ]
 *
730 xfd%=64:yfd%=6:yfc%=8:xfe%=88:y [ 3145 ]
fe%=14:GOSUB 1520
740 ypoint%=9:xpoint%=66:PRINT FNpo [ 5037 ]
s$(7,65);iv$;" coups permis - point
s  ";vi$;
750 REM * Cadre des coups joués *   [ 2101 ]
760 xfd%=64:yfd%=15:yfc%=17:xfe%=88 [ 3249 ]
:yfe%=28:GOSUB 1520
770 ycoupbc%=18:xcoupbc%=66:ycoupnr [ 6616 ]
%=23:xcoupnr%=66:PRINT FNpos$(16,65
);iv$;"   -   coups joués   -    ";vi
$;
780 WHILE (scoreb%+scoren%<>64):nbc [ 3125 ]
oup%=nbcoup%+1
790 PRINT esc$+"f";FNpos$(1,74);iv$ [ 5582 ]
;nbcoup%;vi$;FNpos$(3,78);scoreb%;F
Npos$(4,78);scoren%;
```

```
800 bc%=1:adv%=2:GOSUB 1380                  [ 1567 ]
810 IF couposs%=0 THEN IF indbouc%          [ 5375 ]
THEN 1020:ELSE indbouc%=1:GOTO 910:
ELSE indbouc%=0
820 GOSUB 1590:REM * coups possible         [ 2393 ]
s *
830 PRINT FNpos$(29,0);esc$+"J"+esc         [ 3702 ]
$+"e";"Coup blanc (A1-H8) : ";
840 INPUT;a$:IF a$="" THEN 830:ELSE         [ 8475 ]
 yp%=ASC(LEFT$(a$,1))-64:xp%=ASC(RI
GHT$(a$,1))-48:IF yp%<1 OR yp%>8 TH
EN 830:ELSE IF xp%<1 OR xp%>8 THEN
830
850 IF ecran%(xp%,yp%)<>0 THEN PRIN         [ 7619 ]
T FNpos$(30,0);"coup déjà pris ...
<RC>";:a$=INKEY$:WHILE (a$=""):a$=I
NKEY$:WEND:GOTO 830
860 IF ecrposs%(xp%,yp%)=0 THEN PRI         [ 7673 ]
NT FNpos$(30,0);"coup interdit ...
<RC>";:a$=INKEY$:WHILE (a$=""):a$=I
NKEY$:WEND:GOTO 830
870 PRINT esc$+"f";:REM * plus de c         [ 2659 ]
urseur *
880 GOSUB 1250                              [ 808 ]
890 PRINT FNpos$(1,74);iv$;nbcoup%;         [ 5088 ]
vi$;FNpos$(3,78);scoreb%;FNpos$(4,7
8);scoren%;
900 GOSUB 1650:REM * coups joués bl         [ 2608 ]
ancs *
910 bc%=2:adv%=1:GOSUB 1380                 [ 1569 ]
920 IF couposs%=0 THEN IF indbouc%          [ 5375 ]
THEN 1020:ELSE indbouc%=1:GOTO 800:
ELSE indbouc%=0
930 GOSUB 1590:REM * coups possible         [ 2395 ]
s *
940 IF indjeu%>1 THEN 1730:REM * no         [ 3521 ]
irs jouent seuls ... *
950 PRINT FNpos$(29,0);esc$+"J"+esc         [ 3665 ]
$+"e";"Coup noir (A1-H8) : ";
960 INPUT;a$:IF a$="" THEN 950:ELSE         [ 8487 ]
 yp%=ASC(LEFT$(a$,1))-64:xp%=ASC(RI
GHT$(a$,1))-48:IF yp%<1 OR yp%>8 TH
EN 950:ELSE IF xp%<1 OR xp%>8 THEN
950
970 IF ecran%(xp%,yp%)<>0 THEN PRIN         [ 7625 ]
T FNpos$(30,0);"coup déjà pris ...
<RC>";:a$=INKEY$:WHILE (a$=""):a$=I
NKEY$:WEND:GOTO 950
980 IF ecrposs%(xp%,yp%)=0 THEN PRI         [ 7679 ]
NT FNpos$(30,0);"coup interdit ...
<RC>";:a$=INKEY$:WHILE (a$=""):a$=I
NKEY$:WEND:GOTO 950
990 PRINT esc$+"f";:REM * plus de c         [ 2662 ]
urseur *
1000 GOSUB 1250                             [ 841 ]
1010 GOSUB 1690:REM * coups joués n         [ 2613 ]
oirs *
1020 WEND:PRINT FNpos$(1,74);nbcoup         [ 10706 ]
%;FNpos$(3,78);scoreb%;FNpos$(4,78)
;scoren%;FNpos$(29,0);esc$+"e"+esc$
+"J";"<RC> pour continuer ...";:a$=
INKEY$:WHILE a$="":a$=INKEY$:WEND
1030 IF indjeu%=3 THEN IF scoreb%>=         [ 4723 ]
scorenr% THEN GOSUB 1930:ELSE GOSUB
 1960
1040 GOTO 430                               [ 725 ]
1050 REM * affichage écran *                [ 1707 ]
1060 PRINT cls$;FNpos$(0,10);lig$           [ 1992 ]
1070 PRINT FNpos$(1,10);CHR$(&H96);         [ 8167 ]
:FOR i=0 TO 6:PRINT STRING$(5,CHR$(
&H9A));CHR$(&H9E);:NEXT i:PRINT STR
ING$(5,CHR$(&H9A));CHR$(&H9C);
1080 FOR j=2 TO 24 STEP 3:GOSUB 114         [ 2596 ]
0:NEXT j
1090 PRINT FNpos$(25,10);CHR$(&H93)         [ 8207 ]
;:FOR i=0 TO 6:PRINT STRING$(5,CHR$
(&H9A));CHR$(&H9B);:NEXT i:PRINT ST
RING$(5,CHR$(&H9A));CHR$(&H99);
1100 PRINT FNpos$(26,10);lig$               [ 1722 ]
1110 x%=24:FOR i=1 TO 8:PRINT FNpos         [ 4635 ]
$(x%,7);i;FNpos$(x%,59);i;:x%=x%-3:
NEXT i
1120 RETURN                                 [ 708 ]
1130 REM * une ligne du cadre *             [ 1836 ]
1140 PRINT FNpos$(j,10);CHR$(&H95);         [ 5718 ]
:FOR i=0 TO 7:PRINT STRING$(5,CHR$(
&H20));CHR$(&H95);:NEXT i
1150 PRINT FNpos$(j+1,10);CHR$(&H95         [ 5811 ]
);:FOR i=0 TO 7:PRINT STRING$(5,CHR
$(&H20));CHR$(&H95);:NEXT i
1160 PRINT FNpos$(j+2,10);CHR$(&H97         [ 8288 ]
);:FOR i=0 TO 6:PRINT STRING$(5,CHR
$(&H9A));CHR$(&H9F);:NEXT i:PRINT S
TRING$(5,CHR$(&H9A));CHR$(&H9D);
1170 RETURN                                 [ 713 ]
1180 REM * affichage pion *                 [ 1606 ]
1190 REM * en x%,y% , blanc si bc%=         [ 2966 ]
1, noir si bc%=2 *
1200 xpion%=23-3*(x%-1):ypion%=(y%-         [ 3295 ]
1)*6+12:PRINT CHR$(7);
1210 PRINT FNpos$(xpion%,ypion%);:I         [ 8043 ]
F bc%=1 THEN PRINT CHR$(&HA6);CHR$(
&HAA);CHR$(&HAC);:ELSE PRINT CHR$(&
H96);CHR$(&H9A);CHR$(&H9C);
1220 PRINT FNpos$(xpion%+1,ypion%);         [ 8111 ]
:IF bc%=1 THEN PRINT CHR$(&HA3);CHR
$(&HAB);CHR$(&HA9);:ELSE PRINT CHR$
(&H93);CHR$(&H9A);CHR$(&H99);
1230 ecran%(x%,y%)=bc%                      [ 1235 ]
1240 RETURN                                 [ 711 ]
1250 REM * recherche validité du co         [ 4287 ]
up et calcul points marqués *
1260 gain%=0:FOR sx%=-1 TO 1:FOR sy         [ 4013 ]
%=-1 TO 1:GOSUB 1320:NEXT:NEXT
1270 IF gain%=0 THEN PRINT FNpos$(3         [ 6654 ]
0,0);"coup interdit <RC>";:a$=INKEY
$:WHILE a$="":a$=INKEY$:WEND:RETURN

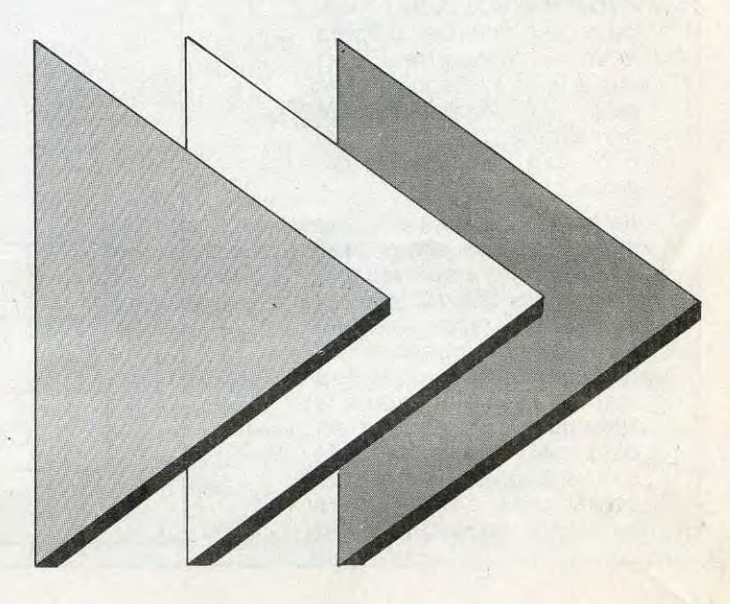
1280 x%=xp%:y%=yp%:GOSUB 1180               [ 1753 ]
1290 IF bc%=1 THEN scoreb%=scoreb%+         [ 8155 ]
gain%+1:scoren%=scoren%-gain%:coupb
c%(icoupbc%)=x%:coupbc%(icoupbc%+1)
=y%:icoupbc%=icoupbc%+2
1300 IF bc%=2 THEN scoreb%=scoreb%-         [ 8310 ]
gain%:scoren%=scoren%+gain%+1:coupn
r%(icoupnr%)=x%:coupnr%(icoupnr%+1)
=y%:icoupnr%=icoupnr%+2
1310 RETURN                                 [ 709 ]
1320 IF ecran%(xp%-sx%,yp%-sy%)<>ad         [ 3071 ]
v% THEN RETURN
1330 i=1:WHILE (ecran%(xp%-i*sx%,yp         [ 3667 ]
%-i*sy%)=adv%):i=i+1:WEND
1340 IF ecran%(xp%-i*sx%,yp%-i*sy%)         [ 3034 ]
=0 THEN RETURN
1350 FOR j=1 TO (i-1):x%=xp%-j*sx%:         [ 4329 ]
y%=yp%-j*sy%:GOSUB 1180:gain%=gain%
+1
1360 NEXT j                                 [ 659 ]
1370 RETURN                                 [ 715 ]
1380 REM * test si au moins un coup         [ 2935 ]
 possible *
1390 FOR i=1 TO 8:FOR j=1 TO 8:ecrp         [ 3527 ]
oss%(i,j)=0:NEXT:NEXT
1400 couposs%=0:FOR xp%=1 TO 8:FOR          [ 5029 ]
yp%=1 TO 8:IF ecran%(xp%,yp%)=0 THE
N GOSUB 1430
1410 NEXT:NEXT:IF couposs%<>0 THEN          [ 2769 ]
RETURN
```

```
1420 PRINT FNpos$(30,0);"aucun coup [ 6722 ]
 n'est possible ... <RC>";:a$=INKEY
$:WHILE (a$=""):a$=INKEY$:WEND:RETU
RN
1430 REM * stockage des coups possi [ 2635 ]
bles *
1440 coup%=0:FOR sx%=-1 TO 1:FOR sy [ 4042 ]
%=-1 TO 1:GOSUB 1460:NEXT:NEXT
1450 RETURN                         [ 714 ]
1460 IF ecran%(xp%-sx%,yp%-sy%)<>ad [ 3076 ]
v% THEN RETURN
1470 i=1:WHILE (ecran%(xp%-i*sx%,yp [ 3672 ]
%-i*sy%)=adv%):i=i+1:WEND
1480 IF ecran%(xp%-i*sx%,yp%-i*sy%) [ 3039 ]
=0 THEN RETURN
1490 FOR j=1 TO (i-1):coup%=coup%+1 [ 2550 ]
:NEXT j
1500 ecrposs%(xp%,yp%)=coup%:coupos [ 3210 ]
s%=couposs%+1:
1510 RETURN                         [ 711 ]
1520 REM * fenetre xfd,yfd xfe,yfe  [ 3058 ]
coupure yfc *
1530 PRINT FNpos$(yfd%,xfd%);CHR$(1 [ 4654 ]
50);STRING$(xfe%-xfd%,CHR$(154));CH
R$(156);
1540 FOR i=1 TO (yfc%-yfd%)-1:PRINT [ 6332 ]
 FNpos$(yfd%+i,xfd%);CHR$(149);FNpo
s$(yfd%+i,xfe%+1);CHR$(149);:NEXT i

1550 IF yfc%<>yfd% THEN PRINT FNpos [ 7379 ]
$(yfc%,xfd%);CHR$(151);:FOR i=1 TO
(xfe%-xfd%):PRINT CHR$(154);:NEXT:P
RINT CHR$(157);
1560 FOR i=1 TO (yfe%-yfc%)-1:PRINT [ 6333 ]
 FNpos$(yfc%+i,xfd%);CHR$(149);FNpo
s$(yfc%+i,xfe%+1);CHR$(149);:NEXT i

1570 PRINT FNpos$(yfe%,xfd%);CHR$(1 [ 4662 ]
47);STRING$(xfe%-xfd%,CHR$(154));CH
R$(153);
1580 RETURN                         [ 718 ]
1590 REM * affichage des coups poss [ 2671 ]
ibles *
1600 ypoint%=9:xpoint%=66           [ 1615 ]
1610 FOR i=0 TO 4:PRINT FNpos$(ypoi [ 4065 ]
nt%+i,66);STRING$(22," ");:NEXT
1620 FOR i=0 TO 8:FOR j=0 TO 8      [ 1748 ]
1630 IF ecrposs%(i,j)<>0 THEN PRINT [ 10607
 FNpos$(ypoint%,xpoint%);CHR$(j+64)
;CHR$(i+48);ecrposs%(i,j)+1;:xpoint
%=xpoint%+6:IF xpoint%>89 THEN xpoi
nt%=66:ypoint%=ypoint%+1
1640 NEXT j:NEXT i:RETURN           [ 1680 ]
1650 REM * affichage des coups blan [ 2887 ]
cs joués *
1660 PRINT FNpos$(ycoupbc%,xcoupbc% [ 6610 ]
);iv$;CHR$(coupbc%(icoupbc%-1)+64);
CHR$(coupbc%(icoupbc%-2)+48);"-";vi
$;
1670 xcoupbc%=xcoupbc%+3:IF xcoupbc [ 4961 ]
%>86 THEN xcoupbc%=66:ycoupbc%=ycou
pbc%+1
1680 RETURN                         [ 719 ]
1690 REM * affichage des coups noir [ 2851 ]
s joués *
1700 PRINT FNpos$(ycoupnr%,xcoupnr% [ 6259 ]
);CHR$(coupnr%(icoupnr%-1)+64);CHR$
(coupnr%(icoupnr%-2)+48);"-";
1710 xcoupnr%=xcoupnr%+3:IF xcoupnr [ 5118 ]
%>86 THEN xcoupnr%=66:ycoupnr%=ycou
pnr%+1
1720 RETURN                         [ 714 ]
1730 REM * recherche de coup pour l [ 2849 ]
es noirs *
1740 PRINT FNpos$(29,0);esc$+"J";"R [ 3562 ]
echerche coup noir ...";
1750 cci%=0:ccj%=0:ccp%=0:ccn%=0    [ 1836 ]
1760 REM * recherche du max de pond [ 2832 ]
ération *
1770 IF icoupnr%>40 THEN 1860:REM * [ 5535 ]
 recherche du couple points,pondéra
tion maximum *
1780 REM * recherche du couple pond [ 3986 ]
ération,points maximum *
1790 FOR i=1 TO 8:FOR j=1 TO 8      [ 1758 ]
1800 IF (ecrposs%(i,j)<>0) AND (che [ 4849 ]
rche%(i-1,j-1)>ccp%) THEN ccp%=cher
che%(i-1,j-1)
1810 NEXT:NEXT                      [ 930 ]
1820 FOR i=1 TO 8:FOR j=1 TO 8      [ 1752 ]
1830 IF (cherche%(i-1,j-1)=ccp%) TH [ 5988 ]
EN IF (ecrposs%(i,j)>ccn%) THEN ccn
%=ecrposs%(i,j):cci%=i:ccj%=j
1840 NEXT:NEXT                      [ 933 ]
1850 xp%=cci%:yp%=ccj%:GOTO 1000    [ 1914 ]
1860 FOR i=1 TO 8:FOR j=1 TO 8      [ 1756 ]
1870 IF ecrposs%(i,j)>ccn% THEN ccn [ 3107 ]
%=ecrposs%(i,j)
1880 NEXT:NEXT                      [ 937 ]
1890 FOR i=1 TO 8:FOR j=1 TO 8      [ 1759 ]
1900 IF ecrposs%(i,j)=ccn% THEN IF  [ 5969 ]
cherche%(i-1,j-1)>ccp% THEN ccp%=ch
erche%(i-1,j-1):cci%=i:ccj%=j
1910 NEXT:NEXT                      [ 931 ]
1920 xp%=cci%:yp%=ccj%:GOTO 1000    [ 1912 ]
1930 REM * report des coups blancs  [ 2266 ]
*
1940 FOR i=0 TO icoupbc%-1 STEP 2:c [ 7786 ]
herche%(coupbc%(i)-1,coupbc%(i+1)-1
)=cherche%(coupbc%(i)-1,coupbc%(i+1
)-1)+1:NEXT i:GOSUB 1990
1950 RETURN                         [ 719 ]
1960 REM * report des coups noirs * [ 2229 ]
1970 FOR i=0 TO icoupnr%-1 STEP 2:c [ 7924 ]
herche%(coupnr%(i)-1,coupnr%(i+1)-1
)=cherche%(coupnr%(i)-1,coupnr%(i+1
)-1)+1:NEXT i:GOSUB 1990
1980 RETURN                         [ 722 ]
1990 REM * écriture de la nouvelle  [ 3229 ]
pondération *
2000 OPEN "O",1,"evol":FOR i=0 TO 7 [ 6054 ]
:FOR j=0 TO 7:WRITE # 1,cherche%(i;
j):NEXT:NEXT:CLOSE 1:RETURN
```

Voir additif technique page suivante...



ADDITIF TECHNIQUE

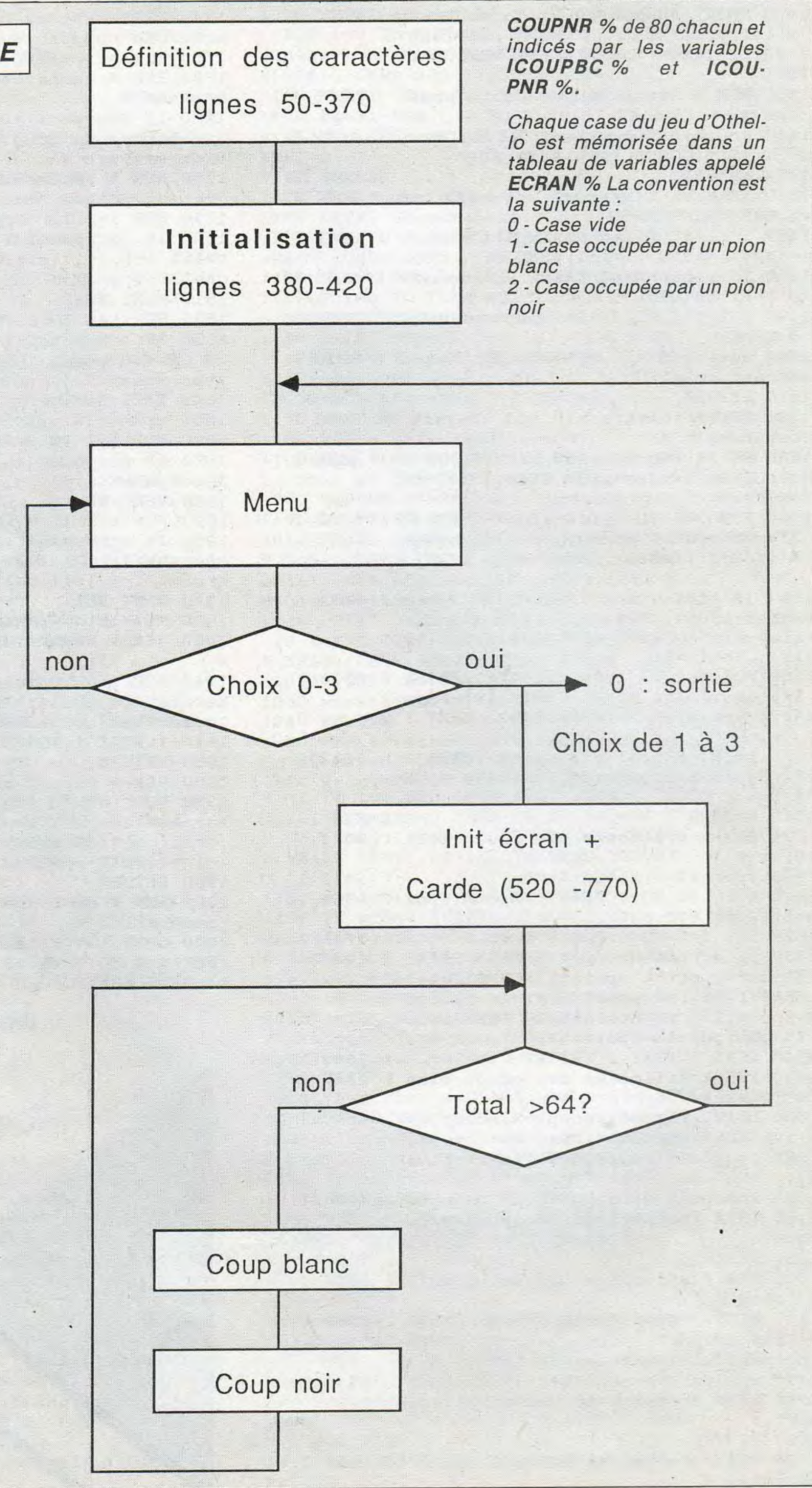
La recherche de validité, ou le calcul du nombre de points marqués lors d'un coup, s'effectue, pour une position donnée, à l'aide d'une « fenêtre » de 8 cases autour de cette position. Ensuite sont explorées, les directions parmi les huit possibles ayant satisfait au test « pion adverse accolé à la position examinée ». Pour éviter les tests de débordement de tableau (assez lourds par le basic), ECRAN % est sur-dimensionné par rapport au jeu :
***Jeu d'Othello :** 8 x 8*
***ECRAN % :** 10 x 10*

Le type de jeu choisi au début est mémorisé dans la variable
INDJEU % :
0 - Fin de jeu (retour vers le basic)
1 - Jeu à deux joueurs
2 - Jeu standard
3 - Jeu évolutif

*La recherche automatique de coups, concernant le joueur aux pions noirs, se fait à l'aide du tableau **CHERCHE % (8 x 8),** contenant une pondération pour chacune des positions du jeu.*
*- Dans le cas du jeu standard **(INDJEU % = 2),** la pondération est fixe et initialisée à partir des dates (lignes 590 à 560).*
*- Dans le cas du jeu évolutif **(INDJEU % = 3),** cette pondération est initialisée d'après le fichier disquette EVOL, mis à jour à chaque fin de partie en incrémentant ladite pondération de 1 pour chaque case jouée par le gagnant.*

*La boucle de jeu s'effectue sur le total de points marqués par les deux joueurs (variables **SCOREN %** et **SCOREB %,** respectivement les scores du joueur noir et du joueur blanc, initialisés à 2 au début du jeu et dont le total maximum ne peut dépasser 64). Les coups joués par les deux adversaires sont mémorisés dans les tableaux **COUPBC %** et **COUPNR %** de 80 chacun et indicés par les variables **ICOUPBC %** et **ICOUPNR %.***

*Chaque case du jeu d'Othello est mémorisée dans un tableau de variables appelé **ECRAN %** La convention est la suivante :*
0 - Case vide
1 - Case occupée par un pion blanc
2 - Case occupée par un pion noir





LA VILLE INFERNALE (PCW)

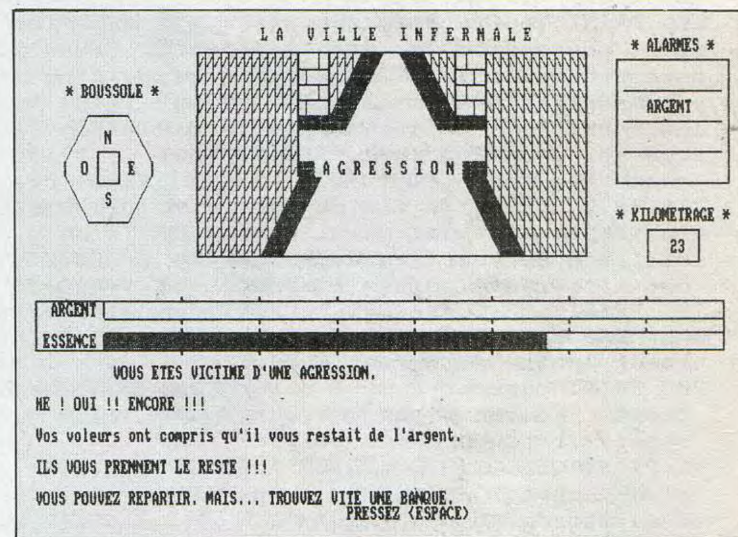
A quoi bon participer au lustre tapageur d'une course folle en milieu exotique, lorsque la traversée d'une ville bien française suffit à satisfaire votre soif d'aventure ? Par ce magnifique programme, vivez les affres d'un malheureux automobiliste, perdu dans un bled paumé de la France profonde..
Soient quatre listings basic à sauver sous les noms suivants :
VILLE.BAS : *lanceur du programme et règle du jeu.*
V1.BAS : *RSX Basic étendu.*
V2. BAS : *mise en place RSX graphique.*
V3.BAS : *programme principal.*

suivi de quatre programmes chargeurs à effacer après usage :
CHARGE1 : *chargeur de RSX Basic étendu.*
CHARGE2 : *chargeur de RSX graphique.*
CHARGE3 : *chargeur de transfert de table de caractères.*
CHARGE4 : *chargeur de la nouvelle table de caractères.*

Le lancement de ces derniers, engendre la création des tois fichiers ; ***VGRAPH. BIN, VCODE.TXT*** *et* ***VJEU. CAR.***

Donc, avant le lancement du programme par ***RUN" VILLE.BAS",*** *vérifiez sur votre disquette l'existence des fichiers :* ***VILLE.BAS, V1.BAS, V2. BAS, V3.BAS, VGRAPH. BIN, VCODE.TXT*** *et* ***VJEU.CAR.*** *Dans ce cas, l'aventure peut commencer.*

Michel Fernandez




```
5 'a sauver sous le nom    VILLE.BAS  [ 2200 ]
6 '                                   [ 125 ]
10 OPTION RUN                         [ 879 ]
20 REM ******* VILLE INFERNALE ****   [ 2408 ]
**********
30 REM copyright FERNANDEZ michel     [ 2255 ]
40 MEMORY &HF605                      [ 972 ]
50 code=HIMEM-400:ad.jeu=code-256*8   [ 2147 ]
60 MEMORY ad.jeu-1                    [ 1140 ]
70 echange.jeu=code+3                 [ 1338 ]
80 x$=CHR$(27):curson$=x$+"e":curso   [ 2693 ]
ff$=x$+"f"
90 cls$=x$+"E"+x$+"H":inv$=x$+"p":n   [ 2608 ]
orm$=x$+"q"
100 DEF FNcu$(li,co)=x$+"Y"+CHR$(32   [ 2902 ]
+li)+CHR$(32+co)
110 DEF FNfe$(li,co,nl,nc)=x$+"X"+C   [ 4623 ]
HR$(32+li)+CHR$(32+co)+CHR$(31+nl)+
CHR$(31+nc)
120 PRINT FNfe$(0,0,32,90);cls$;cur   [ 8802 ]
soff$;FNcu$(0,19);"* *   L A   V I
L L E   I N F E R N A L E   * *":P
RINT FNcu$(2,27);"copiright 1986  F
ERNANDEZ michel"
130 PRINT:PRINT"** REGLES DU JEU **   [ 16968 ]
":PRINT:PRINT"Au volant de votre vo
iture vous vous dirigez vers Paris.
":PRINT"Vous abordez une ville inco
nnue, vous devez la traverser du SU
D au NORD.":PRINT:PRINT"DANS CETTE
VILLE DE 1600 CARREFOURS SONT REPAR
TIS:"
140 PRINT"80 banques; 80 postes de    [ 14318 ]
police, 90 garages, 90 postes d'ess
ence.":PRINT:PRINT"POUR VOUS AIDEZ:
":PRINT"Boussole, Niveaux (essence
et argent)":PRINT"Alarmes (essence,
 argent, consommation, roue de seco
urs)."
150 PRINT:PRINT"POUR VOUS GENER:":P   [ 17050 ]
RINT"Panne de boussole, Panne mécan
ique, Crevaisons, Agressions, Contr
oles de police.":PRINT:PRINT"ARRETS
 POSSIBLES:":PRINT"Banque -> niveau
 argent remis au maximum.":PRINT"Es
sence -> plein du réservoir (à paye
r!)"
160 PRINT"Garage -> remise en état    [ 16568 ]
du véhicule (boussole, roue de seco
urs, carburateur).":PRINT"Poste de
police -> arret obligatoire si pann
eau STOP (controle).":PRINT:PRINT"A
TTENTION La ville n'est pas sure: v
ous pouvez vous faire dévaliser !"
170 PRINT FNfe$(31,0,1,90);FNcu$(0,   [ 2997 ]
40);"PATIENTEZ";
180 OPEN "i",1,"vcode.txt"            [ 1540 ]
190 i=code:WHILE NOT (EOF(1)):INPUT   [ 4931 ]
 #1,octet:POKE i,octet:i=i+1:WEND:C
LOSE #1
200 carad=i:carech=carad+9:POKE car   [ 5923 ]
ech+1,INT(ad.jeu/256):POKE carech,a
```

LA VILLE INFERNALE

```
d.jeu-256*PEEK(carech+1)
210 OPEN "i",1,"vjeu.car"                [ 1437 ]
220 i=-1:FOR t=1 TO 128:INPUT #1,b$      [ 8826 ]
:FOR j=1 TO 31 STEP 2:i=i+1:m$=MID$
(b$,j,2): POKE ad.jeu+i,VAL("&h"+m$
): NEXT j,t:i=i+1:POKE ad.jeu+i,195
:CLOSE #1
230 PRINT cls$;FNcu$(0,36);CHR$(7);      [ 3491 ]
"PRESSEZ UNE TOUCHE";
240 WHILE INKEY$<>"":WEND:WHILE INK      [ 2937 ]
EY$="":WEND
250 PRINT FNfe$(4,0,28,90);cls$;"**      [ 12106 ]
 FIN DU JEU **":PRINT:PRINT"VOUS GA
GNEZ SI vous ressortez de la ville
coté NORD.":PRINT:PRINT"VOUS PERDEZ
 SI:":PRINT"Vous tombez en panne d'
essence."
260 PRINT"Vous n'avez pas assez d'a      [ 15739 ]
rgent pour payer (essence,réparatio
ns,contraventions).":PRINT"Vous ne
vous arretez pas au controle de pol
ice.":PRINT"Vous succombez à la fat
igue (plus de 200 km.).":PRINT"Vous
 ressortez coté SUD."
270 PRINT:PRINT"En fin de partie vo      [ 16689 ]
tre trajet vous est dévoilé avec po
ssibilité de voir les emplacements
 des différents points d'arret.":PR
INT:PRINT"LES COMMANDES:":PRINT"cur
seur gauche = tourner à gauche    c
urseur droite = tourner à droite"
280 PRINT"curseur haut = aller tout      [ 13686 ]
 droit     curseur bas = arret (pour
essence,banque,garage,police)":PRI
NT:PRINT"QUELQUES CONSEILS:":PRINT"
Passez souvent au garage (réparatio
ns garanties pour 10 km.)."
290 PRINT"Surveillez vos alarmes (s      [ 5542 ]
i alarme argent, pas assez pour rép
arations)."
300 PRINT:PRINT"N'hésitez pas à uti      [ 12905 ]
liser papier et crayon (le nord est
 vite perdu).":PRINT"Gardez votre c
alme (les bifurcations sont souvent
 défavorables).":PRINT:PRINT"     B
O N N E   R O U T E"
310 PRINT FNfe$(31,0,1,90);FNcu$(0,      [ 2993 ]
40);"PATIENTEZ";
320 CALL echange.jeu                     [ 1262 ]
330 CLOSE 1                              [ 637 ]
340 CLEAR,62981!:MEMORY 62981!:RUN"      [ 2219 ]
v1
```

```
10 ' La ville infernale                  [ 1445 ]
20 '                                     [ 169 ]
30 ' a sauver sous le nom  V1.BAS        [ 2033 ]
40 '                                     [ 171 ]
50 ' RSX basic etendu                    [ 1328 ]
60 '                                     [ 173 ]
70 MEMORY &HF3FF                         [ 1011 ]
80 DATA C5,D5,23,7E,32,B0,F4,23,7E,      [ 9161 ]
32,B1,F4,3E,80,32,B2,F4,3E,F4,32,B3
,F4,11,B0,F4,0E,98,CD,05,00,11,80,F
4,0E,13,CD,05,00,11,80,F4,0E,16,CD,
05,00,E1,5E,23,56,E1,4E,23,46,D5,C5
90 DATA ED,53,00,F4,ED,43,02,F4,0E,      [ 10201 ]
1A,11,00,F4,CD,05,00,0E,15,11,80,F4
,CD,05,00,E1,29,24,44,D1,C5,D5,0E,1
A,CD,05,00,11,80,F4,0E,15,CD,05,00,
D1,EB,11,80,00,19,EB,C1,10,E7,11,80
,F4,0E,10,CD,05,00,C9
100 DATA D5,23,7E,32,B0,F4,23,7E,32      [ 9192 ]
,B1,F4,3E,80,32,B2,F4,3E,F4,32,B3,F
4,11,B0,F4,0E,98,CD,05,00,11,80,F4,
0E,0F,CD,05,00,FE,00,C0,11,00,F4,0E
,1A,CD,05,00,11,80,F4,0E,14,CD,05,0
0
110 DATA E1,5E,23,56,2A,02,F4,E5,29      [ 9294 ]
,44,D5,78,FE,00,28,18,C5,D5,0E,1A,C
D,05,00,11,80,F4,0E,14,CD,05,00,E1,
11,80,00,19,EB,C1,10,E8,D5,C1,EB,D1
,AF,ED,52,EB,E1,ED,52,C5,E5,11,00,F
4
120 DATA 0E,1A,CD,05,00,11,80,F4,0E      [ 4957 ]
,14,CD,05,00,C1,21,00,F4,D1,ED,B0,1
1,80,F4,0E,10,CD,05,00,C9
130 FOR i=1 TO 119:READ n$:n=VAL("&      [ 3714 ]
h"+n$):POKE &HF4DF+i,n:NEXT
140 FOR i=1 TO 141:READ n$:n=VAL("&      [ 3680 ]
h"+n$):POKE &HF556+i,n:NEXT
150 RUN"v2                               [ 597 ]
```

```
10 ' La ville infernale                  [ 1445 ]
20 '                                     [ 169 ]
30 ' a sauver sous le nom  V2.BAS        [ 2034 ]
40 '                                     [ 171 ]
50 ' RSX graphique                       [ 1167 ]
60 '                                     [ 173 ]
70 bload%=&HF557:adorig=&HF080:dep%      [ 4868 ]
=HIMEM-65536!:dep%=dep%-654:MEMORY
HIMEM-655
80 nom$="VGRAPH.BIN":CALL bload%(no      [ 2668 ]
m$,dep%)
90 DATA F09AF09EF0A4F0A8F0AEF0CDF0E      [ 2396 ]
6F0F5
100 DATA F100F10BF112F119F128F12BF1      [ 2327 ]
7FF19A
110 DATA F1A9F1CCF1F1F1F4F1F9F218F2      [ 2389 ]
1BF220
120 DATA F22DF271F290                    [ 1180 ]
130 DATA F080F0CDF090F095F0D8F0DEF0      [ 3290 ]
F5F100F128F10BF19AF112
140 DATA F080F0D0F091F096F0D8F0DEF0      [ 3290 ]
F8F100F12BF10BF19DF112
150 DIM table(51):m=0:FOR i=1 TO 6:      [ 3390 ]
READ a$:lena=LEN(a$)/4
160 FOR j=1 TO lena:b=VAL("&h"+MID$      [ 3479 ]
(a$,4*j-3,4))-adorig+dep%
170 m=m+1:table(m)=b:NEXT j:NEXT i       [ 2159 ]
180 FOR i=1 TO 27:b=table(i)             [ 1662 ]
190 df=PEEK(b+2)*256+PEEK(b+1)-ador      [ 3432 ]
ig+dep%:df1=FIX(df/256)
200 df2=df-256*df1:POKE b+1,df2:POK      [ 3044 ]
E b+2,df1:NEXT i
210 m=&HF5FF:FOR i=28 TO 39:b=table      [ 3761 ]
(i):b1=FIX(b/256):b2=b-256*b1
220 IF b2<0 THEN b2=b2+256               [ 1439 ]
230 IF b1<0 THEN b1=B1+255               [ 1436 ]
240 POKE m-1,b2:POKE m,b1:m=m-2:NEX      [ 2327 ]
T i
250 m=&HF500:FOR i=40 TO 51:b=table      [ 3709 ]
(i):b1=FIX(b/256):b2=b-256*b1
260 IF b2<0 THEN b2=b2+256               [ 1443 ]
270 IF b1<0 THEN b1=b1+255               [ 1440 ]
280 POKE m,&HC3:POKE m+1,b2:POKE m+      [ 3163 ]
2,b1:m=m+3:NEXT i
290 RUN"v3                               [ 603 ]
```

```
1 ' a sauver sous le nom  V3.BAS    [ 1985 ]
2 '                                 [ 121 ]
3 ' programme principal             [ 1542 ]
4 '                                 [ 123 ]
10 OPTION RUN                       [ 879 ]
20 REM ******* VILLE INFERNALE **** [ 2408 ]
**********
30 REM copyright FERNANDEZ michel   [ 2255 ]
40 DEFINT a-d:DEFINT f-z:DIM vi$(7) [ 2160 ]
50 DEF FNwp(x)=UNT(PEEK(x)+256*PEEK [ 2503 ]
(x+1))
60 gcls=FNwp(&HF5FE):gorigin=FNwp(& [ 3919 ]
HF5FC):gmode=FNwp(&HF5FA)
70 gvideo=FNwp(&HF5F8):gxpos=FNwp(& [ 3942 ]
HF5F6):gypos=FNwp(&HF5F4)
80 gmove=FNwp(&HF5F2):gmover=FNwp(& [ 3934 ]
HF5F0):gplot=FNwp(&HF5EE)
90 gplotr=FNwp(&HF5EC):gdraw=FNwp(& [ 4018 ]
HF5EA):gdrawr=FNwp(&HF5E8)
100 x$=CHR$(27):curson$=x$+"e":curs [ 2734 ]
off$=x$+"f"
110 cls$=x$+"E"+x$+"H":inv$=x$+"p": [ 2649 ]
norm$=x$+"q"
120 DEF FNcu$(li,co)=x$+"Y"+CHR$(32 [ 2904 ]
+li)+CHR$(32+co)
130 DEF FNfe$(li,co,nl,nc)=x$+"X"+C [ 4625 ]
HR$(32+li)+CHR$(32+co)+CHR$(31+nl)+
CHR$(31+nc)
140 z1$=CHR$(170)+CHR$(165)+CHR$(16 [ 9706 ]
5)+CHR$(172):z2$=CHR$(165)+CHR$(165
)+CHR$(171)+CHR$(172):z3$=CHR$(173)
+CHR$(165)+CHR$(171)+CHR$(165):z4$=
CHR$(173)+CHR$(171)+CHR$(165)+CHR$(
165)
150 z5$=CHR$(165)+CHR$(165)+CHR$(17 [ 4930 ]
0):z6$=CHR$(165)+CHR$(171)+CHR$(165
):z7$=" "+CHR$(174)+" "
160 z8$=" "+CHR$(198)+CHR$(254)+" " [ 6636 ]
+CHR$(196)+CHR$(252)+" "+CHR$(196)+
CHR$(197)+" "+CHR$(253)+CHR$(197)+"
 "+CHR$(255)+CHR$(199)+" "
170 GOTO 1090                       [ 731 ]
180 PRINT FNfe$(22,0,11,90);cls$;FN [ 6268 ]
fe$(0,0,32,90);STRING$(12,7);FNcu$(
9,9);" ":PRINT FNfe$(2,21,13,50):RE
TURN
182 PRINT FNfe$(22,0,11,90);cls$;FN [ 5877 ]
fe$(0,0,32,90);CHR$(7);FNcu$(9,9);"
 ":PRINT FNfe$(2,21,13,50):RETURN
185 PRINT FNfe$(22,0,11,90);cls$;FN [ 6594 ]
fe$(0,0,32,90);STRING$(12,7);FNcu$(
9,9);" ":PRINT FNfe$(2,21,13,50);cl
s$:RETURN
190 PRINT STRING$(12,161)+CHR$(184) [ 7032 ]
+STRING$(2,185)+CHR$(187)+"
          "+CHR$(186)+STRING$(2,185
)+CHR$(183)+STRING$(12,160)
200 PRINT STRING$(11,161)+CHR$(184) [ 7086 ]
+STRING$(2,185)+CHR$(187)+"
            "+CHR$(186)+STRING$(2,1
85)+CHR$(183)+STRING$(11,160)
210 PRINT STRING$(10,161)+CHR$(184) [ 7149 ]
+STRING$(2,185)+CHR$(187)+"
              "+CHR$(186)+STRING$(2
,185)+CHR$(183)+STRING$(10,160)
220 PRINT STRING$(9,161)+CHR$(184)+ [ 7134 ]
STRING$(2,185)+CHR$(187)+"
                "+CHR$(186)+STRING$(
2,185)+CHR$(183)+STRING$(9,160)
230 PRINT STRING$(8,161)+CHR$(184)+ [ 7197 ]
STRING$(2,185)+CHR$(187)+"
                  "+CHR$(186)+STRING
$(2,185)+CHR$(183)+STRING$(8,160)
240 RETURN                          [ 662 ]
250 PRINT STRING$(12,161)+CHR$(189) [ 11850 ]
+"  "+CHR$(174)+CHR$(162)+STRING$(3
,161)+CHR$(184)+STRING$(2,185)+CHR$
(187)+"  "+CHR$(186)+STRING$(2,185)
+CHR$(183)+STRING$(3,160)+CHR$(163)
+CHR$(174)+"  "+CHR$(188)+STRING$(1
2,160)
260 PRINT STRING$(12,161)+CHR$(189) [ 9858 ]
+z1$+STRING$(2,161)+CHR$(184)+STRIN
G$(2,185)+CHR$(187)+"      "+CHR$(186
)+STRING$(2,185)+CHR$(183)+STRING$(
2,160)+z3$+CHR$(188)+STRING$(12,160
)
270 PRINT STRING$(12,161)+CHR$(189) [ 9237 ]
+z2$+CHR$(161)+CHR$(184)+STRING$(2,
185)+CHR$(187)+"        "+CHR$(186)+S
TRING$(2,185)+CHR$(183)+CHR$(160)+z
4$+CHR$(188)+STRING$(12,160)
280 PRINT STRING$(12,161)+CHR$(189) [ 8235 ]
+z1$+CHR$(184)+STRING$(2,185)+CHR$(
187)+"          "+CHR$(186)+STRING$(2
,185)+CHR$(183)+z3$+CHR$(188)+STRIN
G$(12,160)
290 PRINT STRING$(12,161)+CHR$(189) [ 6795 ]
+STRING$(6,185)+CHR$(187)+"
 "+CHR$(186)+STRING$(6,185)+CHR$(1
88)+STRING$(12,160)
300 PRINT STRING$(12,161)+CHR$(189) [ 4380 ]
+"                          "+CHR$(18
8)+STRING$(12,160)
310 PRINT STRING$(12,161)+CHR$(189) [ 4381 ]
+"                          "+CHR$(18
8)+STRING$(12,160)
320 PRINT STRING$(12,161)+CHR$(189) [ 6577 ]
+STRING$(3,185)+CHR$(187)+z8$+CHR$(
186)+STRING$(3,185)+CHR$(188)+STRIN
G$(12,160)
330 RETURN                          [ 662 ]
340 PRINT STRING$(20,161)+CHR$(184) [ 6515 ]
+STRING$(2,185)+CHR$(187)+"  "+CHR$
(186)+STRING$(2,185)+CHR$(183)+STRI
NG$(20,160)
350 PRINT STRING$(19,161)+CHR$(184) [ 6596 ]
+STRING$(2,185)+CHR$(187)+"    "+CH
R$(186)+STRING$(2,185)+CHR$(183)+ST
RING$(19,160)
360 PRINT STRING$(18,161)+CHR$(184) [ 6659 ]
+STRING$(2,185)+CHR$(187)+"      "+
CHR$(186)+STRING$(2,185)+CHR$(183)+
STRING$(18,160)
370 PRINT STRING$(17,161)+CHR$(184) [ 6722 ]
+STRING$(2,185)+CHR$(187)+"
"+CHR$(186)+STRING$(2,185)+CHR$(183
)+STRING$(17,160)
380 PRINT STRING$(16,161)+CHR$(184) [ 6785 ]
+STRING$(2,185)+CHR$(187)+"
 "+CHR$(186)+STRING$(2,185)+CHR$(1
83)+STRING$(16,160)
390 PRINT STRING$(15,161)+CHR$(184) [ 6848 ]
+STRING$(2,185)+CHR$(187)+"
  "+CHR$(186)+STRING$(2,185)+CHR$
(183)+STRING$(15,160)
400 PRINT STRING$(14,161)+CHR$(184) [ 6902 ]
+STRING$(2,185)+CHR$(187)+"
    "+CHR$(186)+STRING$(2,185)+CH
R$(183)+STRING$(14,160)
410 PRINT STRING$(13,161)+CHR$(184) [ 6567 ]
+STRING$(2,185)+CHR$(187)+z8$+CHR$(
186)+STRING$(2,185)+CHR$(183)+STRIN
G$(13,160)
```

LA VILLE INFERNALE

```
420 RETURN                              [ 662 ]
430 PRINT STRING$(12,161)+CHR$(189) [ 9183 ]
+"  "+CHR$(174)+CHR$(162)+STRING$(3
,161)+CHR$(184)+STRING$(2,185)+CHR$
(187)+"   "+CHR$(186)+STRING$(2,185)
+CHR$(183)+STRING$(20,160)
440 PRINT STRING$(12,161)+CHR$(189) [ 8227 ]
+z1$+STRING$(2,161)+CHR$(184)+STRIN
G$(2,185)+CHR$(187)+"      "+CHR$(186
)+STRING$(2,185)+CHR$(183)+STRING$(
19,160)
450 PRINT STRING$(12,161)+CHR$(189) [ 7948 ]
+z2$+CHR$(161)+CHR$(184)+STRING$(2,
185)+CHR$(187)+"           "+CHR$(186)+S
TRING$(2,185)+CHR$(183)+STRING$(18,
160)
460 PRINT STRING$(12,161)+CHR$(189) [ 7478 ]
+z1$+CHR$(184)+STRING$(2,185)+CHR$(
187)+"         "+CHR$(186)+STRING$(2
,185)+CHR$(183)+STRING$(17,160)
470 PRINT STRING$(12,161)+CHR$(189) [ 6790 ]
+STRING$(6,185)+CHR$(187)+"
 "+CHR$(186)+STRING$(2,185)+CHR$(1
83)+STRING$(16,160)
480 PRINT STRING$(12,161)+CHR$(189) [ 5618 ]
+"                   "+CHR$(186)+STR
ING$(2,185)+CHR$(183)+STRING$(15,16
0)
490 PRINT STRING$(12,161)+CHR$(189) [ 5650 ]
+"                  "+CHR$(186)+ST
RING$(2,185)+CHR$(183)+STRING$(14,1
60)
500 PRINT STRING$(12,161)+CHR$(189) [ 6572 ]
+STRING$(3,185)+CHR$(187)+z8$+CHR$(
186)+STRING$(2,185)+CHR$(183)+STRIN
G$(13,160)
510 RETURN                              [ 662 ]
520 PRINT STRING$(20,161)+CHR$(184) [ 9182 ]
+STRING$(2,185)+CHR$(187)+"   "+CHR$
(186)+STRING$(2,185)+CHR$(183)+STRI
NG$(3,160)+CHR$(163)+CHR$(174)+"  "
+CHR$(188)+STRING$(12,160)
530 PRINT STRING$(19,161)+CHR$(184) [ 8228 ]
+STRING$(2,185)+CHR$(187)+"     "+CH
R$(186)+STRING$(2,185)+CHR$(183)+ST
RING$(2,160)+z4$+CHR$(188)+STRING$(
12,160)
540 PRINT STRING$(18,161)+CHR$(184) [ 7947 ]
+STRING$(2,185)+CHR$(187)+"         "+
CHR$(186)+STRING$(2,185)+CHR$(183)+
CHR$(160)+z3$+CHR$(188)+STRING$(12,
160)
550 PRINT STRING$(17,161)+CHR$(184) [ 7480 ]
+STRING$(2,185)+CHR$(187)+"
"+CHR$(186)+STRING$(2,185)+CHR$(183
)+z4$+CHR$(188)+STRING$(12,160)
560 PRINT STRING$(16,161)+CHR$(184) [ 6790 ]
+STRING$(2,185)+CHR$(187)+"
 "+CHR$(186)+STRING$(6,185)+CHR$(1
88)+STRING$(12,160)
570 PRINT STRING$(15,161)+CHR$(184) [ 5619 ]
+STRING$(2,185)+CHR$(187)+"
            "+CHR$(188)+STRING$(12,16
0)
580 PRINT STRING$(14,161)+CHR$(184) [ 5651 ]
+STRING$(2,185)+CHR$(187)+"
          "+CHR$(188)+STRING$(12,1
60)
590 PRINT STRING$(13,161)+CHR$(184) [ 6581 ]
+STRING$(2,185)+CHR$(187)+z8$+CHR$(
186)+STRING$(3,185)+CHR$(188)+STRIN
G$(12,160)
600 RETURN                              [ 662 ]
610 PRINT STRING$(12,161)+CHR$(189) [ 5297 ]
+z7$+z7$+z7$+z7$+z7$+z7$+z7$+z7$+CH
R$(188)+STRING$(12,160)
620 PRINT STRING$(12,161)+CHR$(189) [ 5942 ]
+z5$+z5$+z5$+z5$+z5$+z5$+z5$+STRING
$(3,165)+CHR$(188)+STRING$(12,160)
630 PRINT STRING$(12,161)+CHR$(189) [ 5291 ]
+z6$+z6$+z6$+z6$+z6$+z6$+z6$+z6$+CH
R$(188)+STRING$(12,160)
640 PRINT STRING$(12,161)+CHR$(189) [ 5944 ]
+z5$+z5$+z5$+z5$+z5$+z5$+z5$+STRING
$(3,165)+CHR$(188)+STRING$(12,160)
650 PRINT STRING$(12,161)+CHR$(189) [ 4444 ]
+STRING$(24,185)+CHR$(188)+STRING$(
12,160)
660 PRINT STRING$(12,161)+CHR$(189) [ 4389 ]
+"                           "+CHR$(18
8)+STRING$(12,160)
670 PRINT STRING$(12,161)+CHR$(189) [ 4390 ]
+"                           "+CHR$(18
8)+STRING$(12,160)
680 PRINT STRING$(12,161)+CHR$(189) [ 6586 ]
+STRING$(3,185)+CHR$(187)+z8$+CHR$(
186)+STRING$(3,185)+CHR$(188)+STRIN
G$(12,160)
690 RETURN                              [ 671 ]
700 PRINT STRING$(12,161)+CHR$(189) [ 5533 ]
+z7$+z7$+z7$+z7$+z7$+z7$+" "+CHR$(1
74)+CHR$(188)+STRING$(16,160)
710 PRINT STRING$(12,161)+CHR$(189) [ 5916 ]
+z5$+z5$+z5$+z5$+z5$+z5$+CHR$(165)+
CHR$(165)+CHR$(188)+STRING$(16,160)
720 PRINT STRING$(12,161)+CHR$(189) [ 5920 ]
+z6$+z6$+z6$+z6$+z6$+z6$+CHR$(165)+
CHR$(171)+CHR$(188)+STRING$(16,160)
730 PRINT STRING$(12,161)+CHR$(189) [ 5918 ]
+z5$+z5$+z5$+z5$+z5$+z5$+CHR$(165)+
CHR$(165)+CHR$(188)+STRING$(16,160)
740 PRINT STRING$(12,161)+CHR$(189) [ 4439 ]
+STRING$(20,185)+CHR$(183)+STRING$(
16,160)
750 PRINT STRING$(12,161)+CHR$(189) [ 5618 ]
+"                  "+CHR$(186)+STR
ING$(2,185)+CHR$(183)+STRING$(15,16
0)
760 PRINT STRING$(12,161)+CHR$(189) [ 5650 ]
+"                   "+CHR$(186)+ST
RING$(2,185)+CHR$(183)+STRING$(14,1
60)
770 PRINT STRING$(12,161)+CHR$(189) [ 6581 ]
+STRING$(3,185)+CHR$(187)+z8$+CHR$(
186)+STRING$(2,185)+CHR$(183)+STRIN
G$(13,160)
780 RETURN                              [ 671 ]
790 PRINT STRING$(16,161)+CHR$(189) [ 5542 ]
+CHR$(174)+" "+z7$+z7$+z7$+z7$+z7$+
z7$+CHR$(188)+STRING$(12,160)
800 PRINT STRING$(16,161)+CHR$(189) [ 6574 ]
+CHR$(165)+CHR$(172)+z5$+z5$+z5$+z5
$+z5$+STRING$(3,165)+CHR$(188)+STRI
NG$(12,160)
810 PRINT STRING$(16,161)+CHR$(189) [ 5920 ]
+CHR$(171)+CHR$(165)+z6$+z6$+z6$+z6
$+z6$+z6$+CHR$(188)+STRING$(12,160)
820 PRINT STRING$(16,161)+CHR$(189) [ 6576 ]
+CHR$(165)+CHR$(172)+z5$+z5$+z5$+z5
$+z5$+STRING$(3,165)+CHR$(188)+STRI
```

```
NG$(12,160)
830 PRINT STRING$(16,161)+CHR$(184) [ 4439 ]
+STRING$(20,185)+CHR$(188)+STRING$(
12,160)
840 PRINT STRING$(15,161)+CHR$(184) [ 5619 ]
+STRING$(2,185)+CHR$(187)+"
          "+CHR$(188)+STRING$(12,16
0)
850 PRINT STRING$(14,161)+CHR$(184) [ 5651 ]
+STRING$(2,185)+CHR$(187)+"
            "+CHR$(188)+STRING$(12,1
60)
860 PRINT STRING$(13,161)+CHR$(184) [ 6581 ]
+STRING$(2,185)+CHR$(187)+z8$+CHR$(
186)+STRING$(3,185)+CHR$(188)+STRIN
G$(12,160)
870 RETURN                          [ 671 ]
880 PRINT FNcu$(0,0);CHR$(7):PRINT  [ 5400 ]
FNcu$(0,0):PRINT CHR$(161)+CHR$(189
)+z7$+z7$+z7$+CHR$(162)
890 PRINT CHR$(161)+CHR$(189)+z5$+z [ 8301 ]
5$+z5$+CHR$(172):PRINT CHR$(161)+CH
R$(189)+z6$+z6$+z6$+CHR$(172):PRINT
 CHR$(161)+CHR$(189)+z5$+z5$+z5$+CH
R$(172)
900 PRINT CHR$(161)+CHR$(189)+STRIN [ 3413 ]
G$(8,165)+STRING$(2,172)
910 PRINT CHR$(161)+CHR$(189)+nm$+C [ 2950 ]
HR$(162)+CHR$(162)
920 PRINT CHR$(161)+CHR$(189)+" "+C [ 6038 ]
HR$(194)+STRING$(2,190)+CHR$(251)+S
TRING$(2,190)+CHR$(193)+STRING$(2,1
62)
930 PRINT CHR$(161)+CHR$(189)+" "+C [ 4094 ]
HR$(163)+"  "+CHR$(174)+"  "+STRING
$(3,162)
940 PRINT CHR$(161)+CHR$(189)+CHR$( [ 5913 ]
164)+CHR$(177)+STRING$(2,164)+CHR$(
195)+STRING$(2,164)+STRING$(3,178)
950 PRINT CHR$(161)+CHR$(184)+STRIN [ 2577 ]
G$(10,185)
960 PRINT CHR$(184)+STRING$(10,185) [ 4753 ]
:PRINT STRING$(10,185):PRINT STRING
$(9,185)
970 RETURN                          [ 672 ]
980 IF cl=1 THEN RETURN             [ 1468 ]
990 PRINT FNcu$(10,22);CHR$(184)+ST [ 3257 ]
RING$(4,185)+CHR$(183)
1000 PRINT FNcu$(11,22);CHR$(185)+" [ 2844 ]
STOP"+CHR$(185)
1010 PRINT FNcu$(12,22);CHR$(186)+S [ 3297 ]
TRING$(4,185)+CHR$(187)
1020 PRINT FNcu$(13,23);CHR$(176)+" [ 2586 ]
 "+CHR$(175)
1030 RETURN                         [ 708 ]
1040 PRINT CHR$(7);                 [ 1110 ]
1050 WHILE INKEY$<>"":WEND:q$=""    [ 1913 ]
1060 q$=INKEY$:rzz=RND:IF q$=CHR$(1 [ 3102 ]
3)THEN RETURN
1070 IF q$="" THEN 1060             [ 1219 ]
1080 q$=UPPER$(q$):RETURN           [ 1579 ]
1090 PRINT FNfe$(31,0,1,90);cls$;:P [ 8498 ]
RINT FNfe$(0,0,32,90);FNcu$(31,15);
cursoff$;"DEBUT DU VOYAGE";:GOSUB 3
160:GOSUB 1650:GOSUB 2250:GOSUB 226
0
1100 GOSUB 1800                     [ 843 ]
1110 IF y<7 THEN y1=y-1:GOTO 1180   [ 1960 ]
1120 IF y<13 THEN y1=y-7:GOTO 1180  [ 2012 ]
1130 IF y<19 THEN y1=y-13:GOTO 1180 [ 2064 ]
1140 IF y<25 THEN y1=y-19:GOTO 1180 [ 2068 ]
1150 IF y<31 THEN y1=y-25:GOTO 1180 [ 2063 ]
1160 IF y<37 THEN y1=y-31:GOTO 1180 [ 2067 ]
1170 IF y<43 THEN y1=y-37           [ 1466 ]
1180 LI=INT(y/6.1)+1:lo=x+(40*y1)   [ 1903 ]
1190 z=VAL(MID$(vi$(li),lo,1))      [ 1746 ]
1200 GOSUB 2200:GOSUB 2140          [ 1512 ]
1210 IF ar=1 THEN ar=0:GOTO 1225    [ 1888 ]
1220 PRINT FNfe$(2,21,13,50);cls$   [ 1865 ]
1225 PRINT FNfe$(1,21,15,51):ON k G [ 4279 ]
OSUB 250,340,430,520,700,610,790:GO
SUB 190
1230 IF z=0 THEN 1310               [ 1168 ]
1240 ON z GOTO 1250,1260,1270,1280  [ 1825 ]
1250 nm$=" GARAGE ":GOTO 1300       [ 1638 ]
1260 nm$=" ESSENCE":GOTO 1300       [ 1702 ]
1270 nm$=" BANQUE ":GOTO 1300       [ 1661 ]
1280 nm$=" POLICE ":GOTO 1290       [ 1670 ]
1290 GOSUB 880:GOSUB 980:GOTO 1310  [ 2047 ]
1300 GOSUB 880                      [ 804 ]
1310 PRINT FNfe$(22,0,11,90);cls$;F [ 14621 ]
Ncu$(2,36);"QUE FAITES-VOUS ?";FNcu
$(4,36);"tout droit ( ";CHR$(131);"
 )":PRINT FNcu$(5;36);"tourne à dro
ite ( ";CHR$(132);" )";FNcu$(6,36);
"tourne à gauche ( ";CHR$(134);" )"
:PRINT FNcu$(7,36);"arret ( ";CHR$(
133);" )";

1320 WHILE INKEY$<>"":WEND          [ 1609 ]
1330 FF=RND:q$=INKEY$:IF Q$=""THEN  [ 2329 ]
1330
1340 IF ASC(q$)=1 THEN q=1:GOTO 139 [ 2096 ]
0
1350 IF ASC(q$)=6 THEN q=2:GOTO 139 [ 2103 ]
0
1360 IF ASC(q$)=30 THEN q=3:GOTO 13 [ 2150 ]
90
1370 IF ASC(q$)=31 THEN q=4:GOTO 13 [ 2153 ]
90
1380 PRINT CHR$(7);:GOTO 1330       [ 1719 ]
1390 PRINT cls$:IF z=4 AND cl=0 AND [ 2991 ]
 q<>3 THEN 2550
1400 IF (q=1 AND c2=2)OR(q=2 AND c3 [ 3596 ]
=2)OR(q=4 AND c1=2)THEN 2570
1410 IF q=3 THEN 2620               [ 1167 ]
1420 d1=q:cl=0:e=e-cs:km=km+1       [ 1767 ]
1430 IF e<0 THEN 2990               [ 1163 ]
1440 PRINT FNfe$(0,0,32,90):GOSUB 2 [ 2176 ]
260
1450 IF km=200 THEN 3060            [ 1336 ]
1460 IF EN>0 OR km<4 THEN 1520      [ 1755 ]
1470 J=INT(RND*30)+1                [ 1148 ]
1480 IF J=23 AND P<5 THEN 2270      [ 1686 ]
1490 IF J=11 AND P<10 AND AG<2 AND  [ 2734 ]
a>0 THEN 2390
1500 IF (J=5 OR J=16) AND CS=1 AND  [ 2755 ]
P<10 THEN 2460
1510 IF (J=8 OR J=27 OR J=2) AND BL [ 3159 ]
=1 AND P<10 THEN 2510
1520 en=en-1                        [ 681 ]
1530 ON d1 GOTO 1540,1560,2620,1570 [ 1861 ]
1540 D=D-1:IF D=0 THEN D=4          [ 1483 ]
1550 GOTO 1570                      [ 785 ]
1560 D=D+1:IF D=5 THEN D=1          [ 1485 ]
1570 IF D=1 THEN Y=Y-1:F$=f$+"1"    [ 1813 ]
1580 IF D=2 THEN X=X+1:F$=f$+"2"    [ 1812 ]
1590 IF D=3 THEN Y=Y+1:F$=f$+"3"    [ 1817 ]
1600 IF D=4 THEN X=X-1:F$=f$+"4"    [ 1843 ]
1610 P=P-1:IF P<0 THEN P=0          [ 1524 ]
1620 IF Y=0 THEN 3020               [ 1170 ]
1630 IF Y=41 THEN 3100              [ 1223 ]
1640 GOTO 1100                      [ 774 ]
1650 PRINT CURSOFF$;FNfe$(0,0,32,90 [ 5270 ]
```

```
);cls$;STRING$(29,32);"L A   V I L
L E   I N F E R N A L E"
1660 PRINT FNfe$(1,20,16,53);cls$;C [ 10013 ]
HR$(166)+STRING$(50,164)+CHR$(167):
FOR i=0 TO 12:PRINT CHR$(162)+STRIN
G$(50,32)+CHR$(163):NEXT:PRINT CHR$
(169)+STRING$(50,165)+CHR$(168);
1670 PRINT FNfe$(0,0,32,90);FNcu$(1 [ 13749 ]
6,0);STRING$(18,164)+CHR$(129)+STRI
NG$(9,164)+CHR$(129)+STRING$(9,164)
+CHR$(129)+STRING$(9,164)+CHR$(129)
+STRING$(9,164)+CHR$(129)+STRING$(9
,164)+CHR$(129)+STRING$(9,164)+CHR$
(129)+STRING$(10,164);CHR$(167);
1680 PRINT CHR$(163)+" ARGENT"+CHR$ [ 3760 ]
(162)+STRING$(80,32)+CHR$(163);
1690 PRINT CHR$(163)+"          "+STR [ 12947 ]
ING$(9,130)+CHR$(127)+STRING$(9,130
)+CHR$(127)+STRING$(9,130)+CHR$(127
)+STRING$(9,130)+CHR$(127)+STRING$(
9,130)+CHR$(127)+STRING$(9,130)+CHR
$(127)+STRING$(9,130)+CHR$(127)+STR
ING$(10,130)+CHR$(163);
1700 PRINT CHR$(163)+"ESSENCE"+CHR$ [ 3790 ]
(162)+STRING$(80,32)+CHR$(163);
1710 PRINT STRING$(18,165)+CHR$(128 [ 12162 ]
)+STRING$(9,165)+CHR$(128)+STRING$(
9,165)+CHR$(128)+STRING$(9,165)+CHR
$(128)+STRING$(9,165)+CHR$(128)+STR
ING$(9,165)+CHR$(128)+STRING$(9,165
)+CHR$(128)+STRING$(10,165)+CHR$(16
8);
1720 PRINT FNcu$(4,4);"* BOUSSOLE * [ 10624 ]
";FNcu$(6,6);CHR$(191)+STRING$(5,16
5)+CHR$(192);FNcu$(7,5);CHR$(191)+"
   N   "+CHR$(192);FNcu$(8,4);CHR$(
191)+"   "+CHR$(194)+CHR$(190)+CHR$
(193)+"   "+CHR$(192);
1730 PRINT FNcu$(9,3);CHR$(162)+"   [ 7876 ]
O "+CHR$(163)+" "+CHR$(162)+" E  "+
CHR$(163);FNcu$(10,4);CHR$(192)+"
 "+CHR$(175)+CHR$(190)+CHR$(176)+"
 "+CHR$(191)
1740 PRINT FNcu$(11,5);CHR$(192)+"   [ 5397 ]
 S   "+CHR$(191);FNcu$(12,6);CHR$(
192)+STRING$(5,164)+CHR$(191)
1750 PRINT FNcu$(1,77);"* ALARMES * [ 10112 ]
";FNcu$(2,75);CHR$(194)+STRING$(13,
190)+CHR$(193);FNcu$(3,75);CHR$(163
)+STRING$(13,32)+CHR$(162);FNcu$(4,
75);CHR$(163)+STRING$(13,190)+CHR$(
162)
1760 PRINT FNcu$(5,75);CHR$(163)+ST [ 8590 ]
RING$(13,32)+CHR$(162);FNcu$(6,75);
CHR$(163)+STRING$(13,190)+CHR$(162)
;FNCU$(7,75);CHR$(163)+STRING$(13,3
2)+CHR$(162)
1770 PRINT FNcu$(8,75);CHR$(163)+ST [ 8700 ]
RING$(13,190)+CHR$(162);FNcu$(9,75)
;CHR$(163)+STRING$(13,32)+CHR$(162)
;FNcu$(10,75);CHR$(175)+STRING$(13,
190)+CHR$(176)
1780 PRINT FNCU$(12,75);"* KILOMETR [ 9953 ]
AGE *";FNCU$(13,79);CHR$(194)+STRIN
G$(5,190)+CHR$(193);FNCU$(14,79);CH
R$(163)+"    0 "+CHR$(162);FNCU$(15,
79);CHR$(175)+STRING$(5,190)+CHR$(1
76)
1790 RETURN                         [ 721 ]
1800 IF BL=0 THEN 1820              [ 1229 ]
1810 IF KM=0 THEN K=1:RETURN        [ 1760 ]
1820 FF=INT(RND*20)+1               [ 1212 ]
1830 IF FF=4 OR FF=8 OR FF=20 THEN  [ 2827 ]
K=1:GOTO 1900
1840 IF FF=7 OR FF=16 THEN K=2:GOTO [ 2355 ]
 1900
1850 IF FF=1 OR FF=9 THEN K=3:GOTO  [ 2305 ]
1900
1860 IF FF=3 OR FF=14 THEN K=4:GOTO [ 2353 ]
 1900
1870 IF FF=5 OR FF=10 OR FF=13 THEN [ 2879 ]
 K=5:GOTO 1900
1880 IF FF=6 OR FF=12 OR FF=18 THEN [ 2890 ]
 K=7:GOTO 1900
1890 K=6                            [ 432 ]
1900 IF K=1 THEN c1=3:c2=3:c3=3     [ 1764 ]
1910 IF K=2 THEN c1=3:c2=2:c3=2     [ 1764 ]
1920 IF K=3 THEN c1=3:c2=3:c3=2     [ 1767 ]
1930 IF K=4 THEN c1=3:c2=2:c3=3     [ 1769 ]
1940 IF K=5 THEN c1=2:c2=3:c3=2     [ 1770 ]
1950 IF K=6 THEN c1=2:c2=3:c3=3     [ 1773 ]
1960 IF K=7 THEN c1=2:c2=2:c3=3     [ 1774 ]
1970 ON D GOTO 1980,2020,2060,2100  [ 1808 ]
1980 IF X=1 AND C2=3 THEN 1820      [ 1689 ]
1990 IF X=40 AND C3=3 THEN 1820     [ 1742 ]
2000 IF Y=1 AND(X<19 OR X>21) AND C [ 2725 ]
1=3 THEN 1820
2010 RETURN                         [ 707 ]
2020 IF X=40 AND C1=3 THEN 1820     [ 1725 ]
2030 IF Y=1 AND(X<19 OR X>21) AND C [ 2729 ]
2=3 THEN 1820
2040 IF Y=40 AND(X<19 OR X>21) AND  [ 2782 ]
C3=3 THEN 1820
2050 RETURN                         [ 775 ]
2060 IF X=1 AND C3=3 THEN 1820      [ 1680 ]
2070 IF X=40 AND C2=3 THEN 1820     [ 1731 ]
2080 IF Y=40 AND(X<19 OR X>21) AND  [ 2784 ]
C1=3 THEN 1820
2090 RETURN                         [ 715 ]
2100 IF X=1 AND C1=3 THEN 1820      [ 1673 ]
2110 IF Y=1 AND(X<19 OR X>21) AND C [ 2761 ]
3=3 THEN 1820
2120 IF Y=40 AND(X<19 OR X>21) AND  [ 2812 ]
C2=3 THEN 1820
2130 RETURN                         [ 710 ]
2140 IF bl=0 THEN b$="?":GOTO 2200  [ 1917 ]
2150 IF d=1 THEN b$=CHR$(131):GOTO  [ 2209 ]
2190
2160 IF d=2 THEN b$=CHR$(132):GOTO  [ 2212 ]
2190
2170 IF d=3 THEN b$=CHR$(133):GOTO  [ 2215 ]
2190
2180 IF d=4 THEN b$=CHR$(134)       [ 1611 ]
2190 PRINT FNfe$(0,0,32,90);FNcu$(9 [ 2836 ]
,9);b$:RETURN
2200 IF E<10 THEN AL1$="    ESSENCE [ 3257 ]
 " ELSE AL1$="              "
2210 IF A<20 THEN AL2$="     ARGENT [ 3156 ]
 " ELSE AL2$="              "
2220 IF CS=2 THEN AL3$="CONSOMMATIO [ 3475 ]
N " ELSE AL3$="              "
2230 IF RS=0 THEN AL4$="ROUE SECOUR [ 3499 ]
S " ELSE AL4$="              "
2240 PRINT FNfe$(0,0,32,90):PRINT F [ 6343 ]
Ncu$(3,76);al1$;FNcu$(5,76);al2$;FN
cu$(7,76);al3$;FNcu$(9,76);al4$:RET
URN
2250 PRINT FNfe$(0,0,32,90):PRINT F [ 4800 ]
Ncu$(18,9);STRING$(a,185)+SPACE$(80
-a):RETURN
2260 PRINT FNfe$(0,0,32,90):PRINT F [ 8012 ]
Ncu$(20,9);STRING$(e,185)+SPACE$(80
-e):PRINT FNcu$(9,9);b$:PRINT FNcu$
```

```
(14,81);USING"###";km:RETURN
2270 GOSUB 180:PRINT FNcu$(7,15);"  [ 4043 ]
C R E V A I S O N ";FNfe$(22,0,11,9
0);
2280 PRINT"              VOUS VENEZ DE  [ 2565 ]
CREVER !!!"
2290 IF RS<1 THEN 2350              [ 1255 ]
2300 PRINT:PRINT"Heureusement pour  [ 8141 ]
vous votre roue de secours est en b
on état.":PRINT:PRINT"VOUS VOUS DEP
ANNEZ.":rs=0
2310 PRINT:PRINT"Vous pouvez repart [ 9576 ]
ir mais... vous n'avez plus de roue
 de secours.":PRINT:PRINT"FAITES RE
PARER AU PROCHAIN GARAGE.":GOSUB 22
00
2320 PRINT FNfe$(22,0,11,90);FNcu$( [ 5014 ]
10,40);"PRESSEZ <ESPACE>";:GOSUB 10
50:PRINT cls$
2330 EN=1:GOTO 1530                 [ 1093 ]
2340 PRINT FNfe$(22,0,11,90);FNcu$( [ 5614 ]
10,40);"PRESSEZ <ESPACE>";:GOSUB 10
50:PRINT cls$:GOTO 1200
2350 PRINT:PRINT"HE! OUI!! VOUS AVE [ 5983 ]
Z QUATRE ROUES.... ET VOTRE ROUE DE
 SECOURS EST CREVEE."
2360 PRINT:PRINT"Vous devez faire v [ 12682 ]
enir un dépanneur... et cela n'est
pas gratuit.":a=a-20:IF a>0 THEN PR
INT:PRINT"VOUS POUVEZ REPARTIR MAIS
... surveillez votre argent.":GOSUB
 2250:GOTO 2320
2370 GOTO 2720                      [ 784 ]
2390 GOSUB 180:PRINT FNcu$(7,15);"  [ 4047 ]
A G R E S S I O N ";FNfe$(22,0,11,9
0);
2400 PRINT"              VOUS ETES VICT [ 4425 ]
IME D'UNE AGRESSION.":IF AG=1 THEN
2430
2410 PRINT:PRINT"Sous la menace vou [ 5724 ]
s donnez la moitie de votre argent.
":A=INT(A/2):AG=1
2420 PRINT:PRINT"VOUS POUVEZ REPART [ 6183 ]
IR. MAIS... TROUVEZ VITE UNE BANQUE
.":GOSUB 2250:GOTO 2320
2430 PRINT:PRINT"HE ! OUI !! ENCORE [ 2333 ]
 !!!"
2440 PRINT:PRINT"Vos voleurs ont co [ 4974 ]
mpris qu'il vous restait de l'argen
t."
2450 PRINT:PRINT"ILS VOUS PRENNENT  [ 4269 ]
LE RESTE !!!":A=0:AG=2:GOTO 2420
2460 GOSUB 180:PRINT FNcu$(7,9);" P [ 4575 ]
 A N N E   M E C A N I Q U E ";FNfe
$(22,0,11,90);
2470 PRINT"              VOUS AVEZ DES  [ 3324 ]
ENNUIS MECANIQUES !!"
2480 PRINT:PRINT"VOTRE CARBURATEUR  [ 6152 ]
EST DEREGLE. Votre consommation d'e
ssence double.":CS=2
2490 PRINT:PRINT"VOUS POUVEZ REPART [ 5768 ]
IR. MAIS... Pensez a faire régler v
otre voiture."
2500 GOTO 2320                      [ 807 ]
2510 GOSUB 180:PRINT FNcu$(7,16);"  [ 3947 ]
B O U S S O L E ";FNfe$(22,0,11,90)
;
2520 PRINT"              VOTRE BOUSSOLE [ 2788 ]
 SE BLOQUE."
2530 PRINT:PRINT"RIEN NE VOUS INDIQ [ 4610 ]
UE MAINTENANT VOTRE DIRECTION !!!"
2540 PRINT:PRINT"A vous de ne pas p [ 4272 ]
erdre le nord !!":BL=0:GOTO 2490
2550 GOSUB 180:PRINT FNcu$(7,10);"  [ 4469 ]
D E L I T   D E   F U I T E ";FNfe$
(22,0,11,90);
2560 PRINT"VOUS VENEZ DE FORCER UN  [ 10893 ]
BARRAGE DE POLICE...":PRINT:PRINT"C
ETTE ACTION VOUS OBLIGE A TERMINER
VOTRE VOYAGE...":PRINT:PRINT"
    E N   P R I S O N":GOTO 3130
2570 GOSUB 180:PRINT FNcu$(7,10);"  [ 4574 ]
P E R C U S S I O N   M U R ";FNfe$
(22,0,11,90);
2580 PRINT"              VOUS VENEZ DE  [ 3484 ]
PERCUTER UN MUR !!":ar=1
2590 PRINT:PRINT"La réparation est  [ 5578 ]
trés onéreuse. Faites attention que
 diable !!"
2600 PRINT:PRINT"Il faut payer le c [ 10100 ]
arrossier":a=a-30:IF a>0 THEN PRINT
:PRINT"VOUS POUVEZ REPARTIR... mais
 surveillez votre argent":GOSUB 225
0:GOTO 2340
2610 GOTO 2720                      [ 781 ]
2620 GOSUB 182:ar=1:PRINT FNcu$(7,1 [ 4030 ]
8);"  A R R E T   ";FNfe$(22,0,11,9
0);
2630 IF Z>0 THEN 2660               [ 1183 ]
2640 PRINT"              LA RUE EST PAR [ 7876 ]
FAITEMENT DESERTE.":PRINT:PRINT"A p
art vous reposer, rien n'explique c
ette manoeuvre."
2650 PRINT:PRINT"VOUS FERIEZ MIEUX  [ 5931 ]
DE REPARTIR... Vous encombrez la ch
aussée.":GOTO 2340
2660 IF Z>1 THEN 2730               [ 1185 ]
2670 PRINT"VOUS ETES AU GARAGE ...  [ 4549 ]
Votre voiture passe en révision."
2680 PRINT:PRINT"REPARATIONS EFFECT [ 5562 ]
UEES... IL NE VOUS RESTE PLUS QU'A
PAYER.":A=A-20
2690 IF A<0 THEN 2720               [ 1159 ]
2700 PRINT:PRINT"VOUS POUVEZ REPART [ 5002 ]
IR. MAIS... Surveillez votre argent
 !!"
2710 RS=1:P=20:CS=1:BL=1:GOSUB 2250 [ 3383 ]
:GOSUB 2200:GOTO 2340
2720 PRINT:PRINT CHR$(7);"VOUS N'AV [ 9306 ]
EZ PAS ASSEZ D'ARGENT... Vous finis
sez donc votre voyage...":PRINT:PRI
NT"              E N   P R I S
O N":GOTO 3130
2730 IF Z>2 THEN 2810               [ 1183 ]
2740 PRINT"VOUS ETES A LA POMPE A E [ 2752 ]
SSENCE."
2750 IF E=80 THEN 2790              [ 1224 ]
2760 PRINT:PRINT"LE POMPISTE FAIT L [ 5273 ]
E PLEIN... Il ne vous reste plus qu
'a payer."
2770 E=80-E:E=E/2:A=INT(A-E):E=80:I [ 2889 ]
F A<0 THEN 2720
2780 PRINT:PRINT"VOUS POUVEZ REPART [ 7029 ]
IR. MAIS... Surveillez votre argent
 !!":GOSUB 2260:GOSUB 2250:GOTO 234
0
2790 PRINT:PRINT"ARRET INUTILE, RES [ 11724 ]
ERVOIR PLEIN.":PRINT:PRINT"Pour ne
pas avoir l'air ridicule, vous fait
es vérifier vos pneux.":PRINT:PRINT
"Cela vous coute cinq francs."
2800 A=A-5:IF A<0 THEN 2720 ELSE 27 [ 2069 ]
80
```

```
2810 IF Z>3 THEN 2860                    [ 1188 ]
2820 PRINT"VOUS ETES DANS LA BANQUE [ 3427 ]
.":IF A<80 THEN 2840
2830 PRINT:PRINT"VOUS ETES TETE EN  [ 10752 ]
L'AIR... Votre avoir est déjà au ma
ximum.":PRINT:PRINT"VOUS AVEZ L'AIR
 RIDICULE...Vous n'avez plus qu'à r
epartir...":GOTO 2340
2840 PRINT:PRINT"VOUS COMPLETEZ VOT [ 3865 ]
RE AVOIR AU MAXIMUM."
2850 AG=0:A=80:PRINT:PRINT"VOUS POU [ 6732 ]
VEZ REPARTIR. MAIS... Attention aux
 voleurs !!":GOSUB 2250:GOTO 2340
2860 IF cl=1 THEN 2640                   [ 1239 ]
2870 PRINT"CONTROLE DE POLICE... UN [ 5267 ]
 POLICIER EXAMINE VOTRE VOITURE..."
:CL=1:EN=2
2880 IF P<5 THEN 2900                    [ 1180 ]
2890 PRINT:PRINT"VOTRE VOITURE EST  [ 7291 ]
EN BON ETAT.":PRINT:PRINT"Vous pouv
ez reprendre votre route.":GOTO 234
0
2900 FF=INT(RND*11)+1                    [ 1212 ]
2910 IF FF=5 OR FF=11 THEN 2890          [ 1767 ]
2920 IF FF=2 OR FF=6 OR FF=9 THEN 2 [ 2203 ]
970
2930 IF FF=1 OR FF=7 OR FF=10 THEN  [ 2245 ]
2980
2940 PRINT:PRINT"VOS PNEUX SONT EN  [ 3509 ]
MAUVAIS ETAT !!. ";
2950 PRINT"Vous devez payer une con [ 4057 ]
travention.";CHR$(7):A=A-20
2960 IF A<0 THEN 2720 ELSE 2780          [ 1729 ]
2970 PRINT:PRINT"VOTRE ECLAIRAGE ES [ 4201 ]
T DEFFECTUEUX !!. ";:GOTO 2950
2980 PRINT:PRINT"VOTRE VOITURE EST  [ 4242 ]
TROP BRUYANTE !!.";:GOTO 2950
2990 GOSUB 185:PRINT FNcu$(7,9);" P [ 4549 ]
 A N N E   D ' E S S E N C E ";FNfe
$(22,0,11,90);
3000 PRINT"VOUS ETES EN PANNE D'ESS [ 7335 ]
ENCE !!":PRINT:PRINT"DOMMAGE POUR V
OUS MAIS... Votre voyage se termine
 ici."
3010 PRINT:PRINT"La prochaine fois  [ 5592 ]
soyez prévoyant ... PRENEZ LE TRAIN
 !!!!":GOTO 3150
3020 GOSUB 185:GOSUB 185:PRINT FNcu [ 4222 ]
$(7,19);" G A G N E ";FNfe$(22,0,11
,90);
3030 PRINT"B R A V O !!! VOUS ETES  [ 5280 ]
ENFIN PARVENU A TRAVERSER CETTE DRO
LE DE VILLE."
3040 PRINT:PRINT"Vous y avez mis le [ 5624 ]
 temps !!! VOUS AVEZ PARCOURU : ";K
M;" kilomètres"
3050 PRINT:PRINT"VOUS POUVEZ FAIRE  [ 5861 ]
MIEUX... La prochaine fois prenez u
n plan !":GOTO 3150
3060 GOSUB 185:PRINT FNcu$(7,16);"  [ 3904 ]
A C C I D E N T ";FNfe$(22,0,11,90)
;
3070 PRINT"VOUS ETES VICTIME D'UN A [ 6713 ]
CCIDENT...":PRINT:PRINT"APRES 200 K
m PARCOURUS DANS CETTE VILLE..."
3080 PRINT:PRINT"LA FATIGUE VOUS A  [ 5168 ]
SURPRIS.... VOUS FINISSEZ VOTRE VOY
AGE..."
3090 PRINT:PRINT"              A    [ 3269 ]
L ' H O P I T A L":GOTO 3150
3100 GOSUB 185:PRINT FNcu$(7,11);"  [ 4444 ]
R E T O U R   D E P A R T ";FNfe$(2
2,0,11,90);
3110 PRINT"D O M M A G E  !! VOUS V [ 3749 ]
ENEZ DE SORTIR DE LA VILLE"
3120 PRINT:PRINT"A L'ENDROIT OU VOU [ 14092 ]
S ETES ENTRE... Vu votre sens de l'
orientation...":PRINT:PRINT"laissez
 votre voiture et... VOYAGEZ PAR LE
 TRAIN...":PRINT:PRINT"Et sans vous
 tromper de direction...":GOTO 3150
3130 PRINT FNfe$(22,0,11,90);FNcu$( [ 11958 ]
10,40);"PRESSEZ <ESPACE>";:GOSUB 10
40:PRINT FNfe$(2,21,13,50);cls$:PRI
NT:PRINT"               VOUS ETES E
N PRISON":PRINT:PRINT"        IL
FAIT FROID MAIS COURAGE !!!"
3140 PRINT:PRINT:PRINT"        DANS [ 7720 ]
 QUELQUES JOURS VOTRE FAMILLE":PRIN
T:PRINT"       VIENDRA CERTAINEMEN
T VOUS VOIR !!"
3150 PRINT FNfe$(22,0,11,90);FNcu$( [ 4869 ]
10,40);"PRESSEZ <ESPACE>";:GOSUB 10
40:GOTO 3410
3160 PRINT CHR$(7);FNcu$(31,36);"PR [ 4712 ]
ESSEZ UNE TOUCHE";:WHILE INKEY$<>""
:WEND
3170 WHILE INKEY$="":x=RND:WEND         [ 1988 ]
3180 FOR i=1 TO 7:vi$(I)=STRING$(24 [ 2992 ]
0,48):NEXT:f$=""
3190 PRINT cls$:PRINT FNcu$(10,29); [ 4947 ]
"UN INSTANT LA VILLE SE CONSTRUIT";
CHR$(7)
3200 A=80:E=80:D=1:P=20                 [ 1284 ]
3210 K=1:CS=1:BL=1:AG=0:CL=0:KM=0:R [ 2308 ]
S=1
3220 FOR I=1 TO 80                      [ 1008 ]
3230 X=INT(RND*1600)+1:li=INT(x/240 [ 5408 ]
.1)+1:lo=x-(240*(li-1)):IF MID$(vi$
(li),lo,1)<>"0" THEN 3230
3240 IF x=20 OR x=1580 THEN 3230        [ 1802 ]
3250 MID$(vi$(li),lo,1)="3":NEXT i      [ 1948 ]
3260 FOR I=1 TO 80                      [ 1012 ]
3270 X=INT(RND*1600)+1:li=INT(x/240 [ 5416 ]
.1)+1:lo=x-(240*(li-1)):IF MID$(vi$
(li),lo,1)<>"0" THEN 3270
3280 IF x=20 OR x=1580 THEN 3270        [ 1810 ]
3290 MID$(vi$(li),lo,1)="4":NEXT i      [ 1953 ]
3300 PRINT FNcu$(14,21);"ENCORE UN  [ 5202 ]
PEU DE PATIENCE LE BETON N'EST PAS
SEC !!!";CHR$(7)
3310 FOR I=1 TO 90                      [ 1009 ]
3320 X=INT(RND*1600)+1:li=INT(x/240 [ 5408 ]
.1)+1:lo=x-(240*(li-1)):IF MID$(vi$
(li),lo,1)<>"0" THEN 3320
3330 IF x=20 OR x=1580 THEN 3320        [ 1802 ]
3340 MID$(vi$(li),lo,1)="2":NEXT i      [ 1947 ]
3350 FOR I=1 TO 90                      [ 1013 ]
3360 X=INT(RND*1600)+1:li=INT(x/240 [ 5416 ]
.1)+1:lo=x-(240*(li-1)):IF MID$(vi$
(li),lo,1)<>"0" THEN 3360
3370 IF x=20 OR x=1580 THEN 3360        [ 1810 ]
3380 MID$(vi$(li),lo,1)="1":NEXT i      [ 1950 ]
3390 X=20:Y=40                          [ 794 ]
3400 PRINT CHR$(7):RETURN               [ 1591 ]
3410 CALL gcls:PRINT FNfe$(0,0,32,9 [ 3729 ]
0);cls$;cursoff$:GOTO 3450
3420 CALL gdraw(a%,b%):RETURN           [ 1790 ]
3430 CALL gmove(a%,b%):RETURN           [ 1800 ]
3450 a%=0:b%=0:CALL gorigin(a%,b%): [ 2869 ]
CALL gmode(a%)
3460 a%=388:b%=210:GOSUB 3430:a%=53 [ 9982 ]
6:GOSUB 3420:a%=561:GOSUB 3430:a%=7
```

```
16:GOSUB 3420:b%=46:GOSUB 3420:a%=5
61:GOSUB 3420:a%=536:GOSUB 3430:a%=
388:GOSUB 3420:b%=210:GOSUB 3420
3470 PRINT FNcu$(1,26);"* *   V O T [ 10921 ]
 R E   P A R C O U R S    * *";FNcu$
(4,65);"sortie";FNcu$(27,65);"entré
e";FNcu$(4,1);"DISTANCE PARCOURUE "
;USING"###";km:PRINT FNcu$(4,24);"k
ilomètres"
3472 PRINT FNcu$(7,5);"POSITIONNEME [ 11913 ]
NT":PRINT:PRINT"      E -> essences"
:PRINT:PRINT"      G -> garages":PRI
NT:PRINT"      B -> banques":PRINT:P
RINT"      P -> polices":PRINT:PRINT
"      T -> total"
3475 ff=500:GOSUB 3480:GOTO 3560     [ 1883 ]
3480 a%=549:b%=44:GOSUB 3430:b%=50:  [ 2564 ]
GOSUB 3420
3490 a%=549:b%=50:FOR i=1 TO LEN(f$  [ 3181 ]
):j=VAL(MID$(f$,i,1))
3500 ON j GOTO 3510,3520,3530,3540   [ 1814 ]
3510 b%=B%+4:GOTO 3550               [ 1203 ]
3520 a%=a%+8:GOTO 3550               [ 1206 ]
3530 b%=b%-4:GOTO 3550               [ 1207 ]
3540 a%=a%-8                         [ 602 ]
3550 GOSUB 3420:FOR t=1 TO ff:NEXT   [ 7621 ]
t:NEXT i:a%=a%-1:GOSUB 3430:a%=a%+3
:GOSUB 3420:b%=b%+1:GOSUB 3430:a%=a
%-3:GOSUB 3420:RETURN
3560 PRINT FNfe$(24,0,11,46);cls$;"  [ 5262 ]
VOULEZ-VOUS UN POSITIONNEMENT ? (O/
N)":GOSUB 1040
3570 IF q$<>"O"THEN  RUN             [ 1412 ]
3580 PRINT cls$;"LEQUEL ? ( E-G-B-P  [ 3007 ]
-T )":GOSUB 1040
3590 IF q$="E"THEN a$="2":GOTO 3650  [ 1955 ]
3600 IF q$="G"THEN a$="1":GOTO 3650  [ 1948 ]
3610 IF q$="B"THEN a$="3":GOTO 3650  [ 1946 ]
3620 IF q$="P"THEN a$="4":GOTO 3650  [ 1962 ]
3622 IF q$="T"THEN a$="5":GOTO 3650  [ 2001 ]

3624 GOTO 3560                       [ 790 ]
3630 IF a$="5"THEN 3635              [ 1238 ]
3632 IF MID$(vi$(j),d,1)=a$ THEN GO  [ 3834 ]
SUB 3642:RETURN ELSE RETURN
3635 IF MID$(vi$(j),d,1)<>"0" THEN   [ 3913 ]
GOSUB 3642:RETURN ELSE RETURN
3642 CALL gmove(a%,b%):a%=a%+3:CALL  [ 3299 ]
 gdraw(a%,b%):RETURN
3650 PRINT cls$;Q$;FNfe$(6,49,20,40  [ 3371 ]
);cls$:ff=1:GOSUB 3480
3652 b%=205:FOR j=1 TO 7             [ 1342 ]
3660 FOR i=1 TO 40:a%=387+(i*8):d=i  [ 2628 ]
:GOSUB 3630
3662 NEXT:b%=b%-4                    [ 982 ]
3670 FOR i=1 TO 40:a%=387+(i*8):d=i  [ 2772 ]
+40:GOSUB 3630
3672 NEXT:b%=b%-4                    [ 983 ]
3680 FOR i=1 TO 40:a%=387+(i*8):d=i  [ 2777 ]
+80:GOSUB 3630
3682 NEXT:b%=b%-4                    [ 984 ]
3690 FOR i=1 TO 40:a%=387+(i*8):d=i  [ 2821 ]
+120:GOSUB 3630
3692 NEXT:b%=b%-4                    [ 985 ]
3700 FOR i=1 TO 40:a%=387+(i*8):d=i  [ 2817 ]
+160:GOSUB 3630
3702 NEXT:b%=b%-4                    [ 977 ]
3710 FOR i=1 TO 40:a%=387+(i*8):d=i  [ 2813 ]
+200:GOSUB 3630
3712 NEXT:b%=b%-4:NEXT j:GOTO 3560   [ 2070 ]
```

```
10 ' La ville infernale - chargeur  [ 2228 ]
1
20 '                                [ 169 ]
30 ' CHARGEUR 1 - chargeur de RSX b [ 2903 ]
asic etendu
40 '                                [ 171 ]
50 MEMORY &HF3FF                    [ 1009 ]
60 DATA C5,D5,23,7E,32,B0,F4,23,7E, [ 9159 ]
32,B1,F4,3E,80,32,B2,F4,3E,F4,32,B3
,F4,11,B0,F4,0E,98,CD,05,00,11,80,F
4,0E,13,CD,05,00,11,80,F4,0E,16,CD,
05,00,E1,5E,23,56,E1,4E,23,46,D5,C5

70 DATA ED,53,00,F4,ED,43,02,F4,0E, [ 10199 ]
1A,11,00,F4,CD,05,00,0E,15,11,80,F4
,CD,05,00,E1,29,24,44,D1,C5,D5,0E,1
A,CD,05,00,11,80,F4,0E,15,CD,05,00,
D1,EB,11,80,00,19,EB,C1,10,E7,11,80
,F4,0E,10,CD,05,00,C9
80 DATA D5,23,7E,32,B0,F4,23,7E,32, [ 9151 ]
B1,F4,3E,80,32,B2,F4,3E,F4,32,B3,F4
,11,B0,F4,0E,98,CD,05,00,11,80,F4,0
E,0F,CD,05,00,FE,00,C0,11,00,F4,0E,
1A,CD,05,00,11,80,F4,0E,14,CD,05,00

90 DATA E1,5E,23,56,2A,02,F4,E5,29, [ 9253 ]
44,D5,78,FE,00,28,18,C5,D5,0E,1A,CD
,05,00,11,80,F4,0E,14,CD,05,00,E1,1
1,80,00,19,EB,C1,10,E8,D5,C1,EB,D1,
AF,ED,52,EB,E1,ED,52,C5,E5,11,00,F4

100 DATA 0E,1A,CD,05,00,11,80,F4,0E [ 4955 ]
,14,CD,05,00,C1,21,00,F4,D1,ED,B0,1
1,80,F4,0E,10,CD,05,00,C9
110 FOR i=1 TO 119:READ n$:n=VAL("& [ 3712 ]
h"+n$):POKE &HF4DF+i,n:NEXT
120 FOR i=1 TO 141:READ n$:n=VAL("& [ 3678 ]
h"+n$):POKE &HF556+i,n:NEXT
130 DELETE 50-130                   [ 941 ]
```

```
10 ' La ville infernale - chargeur  [ 2229 ]
2
20 '                                [ 169 ]
30 ' CHARGEUR 2 - a effacer apres l [ 2681 ]
ancement
40 '                                [ 171 ]
50 ' chargeur de RSX graphique      [ 1961 ]
60 '                                [ 173 ]
70 ' ATTENTION ce chargeur ne fonct [ 3275 ]
ionne qu'apres
80 ' avoir lancé le RSX basic étend [ 3094 ]
u (vcharge1)
90 '                                [ 176 ]
100 MEMORY &HF07F:bsave%=&HF4E0:nom [ 3085 ]
$="VGRAPH.BIN"
110 DATA 07CF079406AA07DC07E609460A [ 2298 ]
1C0A36
120 DATA 09B00B48092A0974053805E006 [ 2235 ]
A6054E
130 DATA 07170B420B5606B20B10062D09 [ 2221 ]
7407F1
140 DATA 092E097E0A180AFE0A12080508 [ 2272 ]
6A0886
150 DATA 0A7F098209EC0A61086F04FC04 [ 2270 ]
110553
```

```
160 DATA 0623                         [ 700 ]
170 DATA CD5AFCC200AF2180F506307723 [ 2332 ]
10FCC9
180 DATA 7E3280F5C97E3D200AAF32C2F1 [ 2356 ]
3C32BF
190 DATA F11808AF32BFF13C32C2F10102 [ 2304 ]
0621BD
200 DATA F15EE5C5CD0500C1E12310F5C9 [ 2302 ]
1B6300
210 DATA 1B62004E2346C5EB5E2356E1C9 [ 2335 ]
CDC3F1
220 DATA 2286F5ED5388F5C9ED5B82F518 [ 2373 ]
04ED5B
230 DATA 84F5732372C9CDC3F1E52A88F5 [ 2363 ]
19EBC1
240 DATA 2A86F509C9CDE6F12282F5ED53 [ 2375 ]
84F5C9
250 DATA CD19F22282F5ED5384F5C9CD19 [ 2384 ]
F2CD18
260 DATA F5C9CD19F2CD1EF5C9CDC3F1E5 [ 2446 ]
2A84F5
270 DATA 19EBC12A82F509C9CDE6F10134 [ 2379 ]
F2CD5A
280 DATA FCE900C9E57B2F5FD5E5CB3FCB [ 2474 ]
3FCB3F
290 DATA 5F21005A06082930011910FAEB [ 2288 ]
E1CB3C
300 DATA CB1DCB3CCB1DCB3D190603CB25 [ 2370 ]
CB1410
310 DATA FA11305919D17BE60716005F19 [ 2295 ]
D17BE6
320 DATA 0720043E801807473E80CB3F10 [ 2304 ]
FC5ECD
330 DATA 84F277C9573A80F5FE00280BFE [ 2305 ]
012803
340 DATA 7AABC97A2FA3C97AB3C9CDE6F1 [ 2468 ]
EBD5E5
350 DATA E52A82F5B7ED52E5C1FABCF22A [ 2430 ]
82F5EB
360 DATA 2282F52A84F5E32284F5180821 [ 2246 ]
0000B7
370 DATA ED42E5C1D12A84F5B7ED52EBF2 [ 2393 ]
E1F221
380 DATA 0000B7ED52E5D1B7ED42210100 [ 2249 ]
303118
390 DATA 0AD5E1B7ED4221FFFF301322B0 [ 2366 ]
F4C5E1
400 DATA AF32ABF332B8F33E2332C2F318 [ 2332 ]
26E52A
410 DATA 82F5092282F52A84F5B7ED5222 [ 2314 ]
84F5E1
420 DATA 22B0F4C5E1EB3EEB32ABF332B8 [ 2377 ]
F33E13
430 DATA 32C2F3AF32B2F413ED53B4F423 [ 2382 ]
CDCCF3
440 DATA 22B6F4ED53B8F4ED4BB4F4C5D1 [ 2439 ]
CB3ACB
450 DATA 1BC5ED4BB6F42AB8F419EB2AB4 [ 2444 ]
F4B7ED
460 DATA 52300719EBB7ED52EB03D53AB2 [ 2343 ]
F4B728
470 DATA 1D2A82F5E5D1092282F5E5C10B [ 2333 ]
2A84F5
480 DATA E5CDABF3D12AB0F4192284F518 [ 2361 ]
1D2A84
490 DATA F5E5D1092284F5E5C10BEBED5B [ 2394 ]
82F5D5
500 DATA CDABF3D12AB0F4192282F5D1C1 [ 2358 ]
0B78B1
510 DATA 209FE12284F5E12282F5C9EBE5 [ 2332 ]
D56960
520 DATA B7ED524D4403E1D1EBC5D5E5CD [ 2395 ]
18F5E1
530 DATA D1C1130BAFB920F1B820EEC9C5 [ 2356 ]
EB0601
540 DATA 7CB720097ABD3805652E000609 [ 2287 ]
7B957A
550 DATA 9C3805042930F63F3F78444D21 [ 2249 ]
00003D
560 DATA 2003181729F5781F47791F4F7B [ 2287 ]
917A98
570 DATA 3805577B915F2CF13D20E937C1 [ 2093 ]
C9
580 DATA F09AF09EF0A4F0A8F0AEF0CDF0 [ 2448 ]
E6F0F5
590 DATA F100F10BF112F119F128F12BF1 [ 2340 ]
7FF19A
600 DATA F1A9F1CCF1F1F1F4F1F9F218F2 [ 2393 ]
1BF220
610 DATA F22DF271F290                 [ 1184 ]
620 DIM test%(41):m=0:FOR i=1 TO 6: [ 3388 ]
READ a$:lena=LEN(a$)/4
630 FOR j=1 TO lena:b=VAL("&h"+MID$ [ 4902 ]
(a$,4*(j-1)+1,4)):m=m+1:test%(m)=b:
NEXT j:NEXT i
640 m=&HF080-1:FOR i=1 TO 41:READ a [ 3380 ]
$:lena=LEN(a$)/2:tot=0
650 FOR j=1 TO lena:b=VAL("&h"+MID$ [ 4847 ]
(a$,2*j-1,2)):m=m+1:tot=tot+b:POKE
m,b:NEXT j
660 IF tot<>test%(i) THEN PRINT" Er [ 7319 ]
reur en ligne ",160+10*i," ou en li
gne ",110+10*FIX((i-1)/8),HEX$(tot,
4),HEX$(test%(i),4)
670 NEXT i                            [ 613 ]
680 FOR i=1 TO 4:READ a$:lena=LEN(a [ 2301 ]
$)/4
690 FOR j=1 TO lena:b=VAL("&h"+MID$ [ 2707 ]
(a$,4*j-3,4))
700 POKE b+2,(PEEK(b+2)-1):NEXT j:N [ 2394 ]
EXT i
710 dep%=&HF080:lon%=654:CALL bsave [ 3154 ]
%(nom$,dep%,lon%)
```

```
10 ' La ville infernale - chargeur  [ 2230 ]
3
20 '                                  [ 169 ]
30 ' CHARGEUR 3 - a effacer aprés l [ 2736 ]
ancement
40 '                                  [ 171 ]
50 'chargeur pour transfert de tabl [ 3403 ]
e de caractères
60 '                                  [ 173 ]
70 OPEN "O",1,"VCODE.TXT"            [ 1496 ]
80 FOR i=1 TO 11:READ a$:FOR j=1 TO [ 5285 ]
 31 STEP 2:a=VAL("&H"+MID$(a$,j,2))
:PRINT #1,a:NEXT j,i
90 PRINT #1,16,238,201:CLOSE #1      [ 1737 ]
100 DATA C395F4C3B2F4C3E7F4DD210100 [ 2344 ]
DD6E00
110 DATA DD660111570019113000010300 [ 2201 ]
EDB0C9
120 DATA CD7EF4019FF4F7E900C9DD2131 [ 2407 ]
F5DD5E
130 DATA 00DD56012100B8010008EDB0C9 [ 2300 ]
CD7EF4
140 DATA 01BCF4F7E900C9DD2131F5DD5E [ 2373 ]
00DD56
```

```
150 DATA 01D5DDE1FD2100B8010008FD66 [ 2321 ]
00DD6E
160 DATA 00DD7400FD7500DD23FD230B3E [ 2344 ]
FFB820
170 DATA EAC9CD7EF401F1F4F7E900C9DD [ 2410 ]
2128F5
180 DATA DD6E002600292929EB2100B819 [ 2309 ]
E5DDE1
190 DATA FD2131F5FD6E00FD660119E5FD [ 2350 ]
E10608
200 DATA DD6E00FD6600FD7500DD7400DD [ 2346 ]
23FD23
```

```
10 ' La ville infernale - chargeur  [ 2231 ]
4
20 '                                [ 169 ]
30 ' ce chargeur est à effacer apré [ 3201 ]
s utilisation
40 '                                [ 171 ]
50 'chargeur 4 -  pour nouvelle tab [ 3348 ]
le de caractères
60 '                                [ 173 ]
70 OPEN "O",1,"VJEU.CAR"            [ 1399 ]
80 FOR i=1 TO 128:READ b$:PRINT #1, [ 2157 ]
b$
90 NEXT i:PRINT #1,"C3":CLOSE #1    [ 1910 ]
100 DATA 000066DBDBDB66003C42819981 [ 2257 ]
423C00
110 DATA FEC6C0C0C0C0001038386C6C [ 2332 ]
C6FE00
120 DATA 3C42A599A5423C000000002418 [ 2168 ]
240000
130 DATA 000018007E0018000000181800 [ 2118 ]
666600
140 DATA FE6C6C6C6C6C00181818187E [ 2319 ]
3C1800
150 DATA FEC6603060C6FE00002060FEFE [ 2285 ]
602000
160 DATA 00080CFEFE0C08000018187E18 [ 2265 ]
187E00
170 DATA 002466FFFF6624007CC6C6C6C6 [ 2351 ]
6CEE00
180 DATA 000073DECCDE73007CC6C6FCC6 [ 2391 ]
C6F8C0
190 DATA 66663C6666663C003C603C6666 [ 2248 ]
663C00
200 DATA 00001E307C301E00386CC6FEC6 [ 2269 ]
6C3800
210 DATA 00C06030386CC6000000666666 [ 2192 ]
7C6060
220 DATA 0000FE6C6C6C6C0000003C6666 [ 2257 ]
7C6060
230 DATA 00007ED8D8D870000073CE1818 [ 2246 ]
180C00
240 DATA 03063C66663C60C000E63C1838 [ 2254 ]
6CC700
250 DATA 0306666663C60C0000066C3DB [ 2263 ]
DB7E00
260 DATA 00000000000000001818181818 [ 2088 ]
001800
270 DATA 6C6C6C00000000006C6CFE6CFE [ 2321 ]
6C6C00
280 DATA 183E583C1A7C180000C6CC1830 [ 2269 ]
66C600
290 DATA 386C3876DCCC76001818180000 [ 2214 ]
000000
300 DATA 0C18303030180C0030180C0C0C [ 2175 ]
183000
310 DATA 00663CFF3C6600000018187E18 [ 2206 ]
180000
320 DATA 00000000001818300000007E00 [ 2080 ]
000000
330 DATA 000000000018180006 0C183060 [ 2120 ]
C08000
340 DATA 386CCED6E66C38001838181818 [ 2282 ]
187E00
350 DATA 3C66063C60667E003C66061C06 [ 2242 ]
663C00
360 DATA 1C3C6CCCFE0C1E007E62607C06 [ 2332 ]
663C00
370 DATA 3C66607C66663C007E66060C18 [ 2240 ]
181800
380 DATA 3C66663C66663C003C66663E06 [ 2261 ]
663C00
390 DATA 00001818001818000000181800 [ 2113 ]
181830
400 DATA 0C18306030180C0000007E007E [ 2154 ]
000000
410 DATA 6030180C183060003C66660C18 [ 2169 ]
001800
420 DATA 300C780C7CCC7600183C66667E [ 2272 ]
666600
430 DATA FC66667C6666FC003C66C0C0C0 [ 2314 ]
663C00
440 DATA F86C6666666CF800FE62687868 [ 2325 ]
62FE00
450 DATA FE6268786860F0003C66C0C0CE [ 2305 ]
663E00
460 DATA 6666667E666666007E18181818 [ 2237 ]
187E00
470 DATA 1E0C0C0CCCCC7800E6666C786C [ 2350 ]
66E600
480 DATA F06060606266FE00C6EEFEFED6 [ 2370 ]
C6C600
490 DATA C6E6F6DECEC6C6007CC6C6C6C6 [ 2427 ]
C67C00
500 DATA FC66667C6060F0007CC6C6C6DA [ 2345 ]
CC7600
510 DATA FC66667C6C66E6003C66603C06 [ 2289 ]
663C00
520 DATA 7E5A181818183C006666666666 [ 2235 ]
663C00
530 DATA 666666666663C1800C6C6C6D6FE [ 2336 ]
EEC600
540 DATA C66C38386CC6C6006666663C18 [ 2280 ]
183C00
550 DATA FEC68C183266FE003844380000 [ 2230 ]
000000
560 DATA 00003C60603C00181E30386C38 [ 2205 ]
18F000
570 DATA 183C7E18181818000000000000 [ 2177 ]
0000FF
580 DATA 30180C0000000000000780C7C [ 2181 ]
CC7600
590 DATA E0607C666666DC0000003C6660 [ 2242 ]
663C00
600 DATA 1C0C7CCCCCCC76000003C667E [ 2314 ]
603C00
610 DATA 1C3630783030780000003E6666 [ 2205 ]
3E067C
620 DATA E0607C666666E6001800381818 [ 2219 ]
183C00
630 DATA 06000E060666663CE060666C78 [ 2251 ]
6CE600
640 DATA 38181818181 83C0000006CFED6 [ 2259 ]
D6C600
650 DATA 0000DC666666660000003C6666 [ 2204 ]
663C00
```

```
660 DATA 0000DC66667C60F0000076CCCC [ 2311 ]
7C0C1E
670 DATA 0000DC766060F00000003C603C [ 2207 ]
067C00
680 DATA 30307C3030361C000000666666 [ 2179 ]
663E00
690 DATA 00006666663C18000000C6D6D6 [ 2253 ]
FE6C00
700 DATA 0000C66C386CC6000000666666 [ 2236 ]
3E067C
710 DATA 00007E4C18327E000C303C667E [ 2239 ]
603C00
720 DATA 300C006666663C00300C3C667E [ 2227 ]
603C00
730 DATA 006600000000000000FF03030000 [ 2148 ]
0303FF
740 DATA FF0303000000000000000000000 [ 2137 ]
0303FF
750 DATA FF000000000000FF183C7EFF18 [ 2256 ]
181800
760 DATA 000C0EFFFF0E0C00181818FF7E [ 2337 ]
3C1800
770 DATA 003070FFFF7030002424272020 [ 2194 ]
272424
780 DATA 242424E4E42424242424E40404 [ 2213 ]
FC0000
790 DATA 0000FF0000FF00002424E70000 [ 2214 ]
FF0000
800 DATA 0000FC0404E424242424E40404 [ 2202 ]
E42424
810 DATA 0000FF0000E724242424E70000 [ 2199 ]
E72424
820 DATA 00000018180000001818181818 [ 2099 ]
000000
830 DATA 0000001F1F000000181818F0F [ 2155 ]
000000
840 DATA 00000018181818181818181818 [ 2155 ]
181818
850 DATA 0000000F1F18181818181818F1F [ 2211 ]
181818
860 DATA 000000F8F800000018181818F8F0 [ 2179 ]
000000
870 DATA 000000FFFF000000181818FFFF [ 2244 ]
000000
880 DATA 000000F0F818181818181818F8F8 [ 2235 ]
181818
890 DATA 000000FFFF181818181818FFFF [ 2300 ]
181818
900 DATA 8080A09088848280010105091 1 [ 2150 ]
214101
910 DATA 03030303030303C0C0C0C0C0 [ 2212 ]
C0C0C0
920 DATA 00000000000000FFFF00000000 [ 2125 ]
000000
930 DATA 00000000000000030000000000 [ 2060 ]
0000C0
940 DATA C0000000000000000300000000 [ 2061 ]
000000
950 DATA FF01010101010101FF08080808 [ 2191 ]
080808
960 DATA FF03030303030303FFC0C0C0C0 [ 2283 ]
C0C0C0
970 DATA 08080808080808C0C0C0FF00 [ 2207 ]
000000
980 DATA 030303FF00000000C0C0C0C0C0 [ 2273 ]
C0C0FF
990 DATA 0303030303030303FFFF80808080 [ 2209 ]
808080
1000 DATA FFFFFFFFFFFFFFFFFF1818181 [ 2534 ]
8181818
1010 DATA 181818181818FF8040A050A [ 2295 ]
854AA55
1020 DATA 000102050A152A55AA55AA55A [ 2313 ]
A55AA55
1030 DATA AA552A150A050201AA54A850A [ 2274 ]
0408000
1040 DATA C0C0C0D0C8C4C2C00303030B1 [ 2292 ]
3234303
1050 DATA 000000FF0000000010204081 [ 2154 ]
0204080
1060 DATA 8040201008040201000000FF0 [ 2167 ]
3030303
1070 DATA 000000FFC0C0C0C0080808080 [ 2302 ]
80808FF
1080 DATA 030307070F0F1F1F8080C0C0E [ 2333 ]
0E0F0F0
1090 DATA 0103070F1F3F7FFF80C0E0F0F [ 2466 ]
8FCFEFF
1100 DATA 18663C6666663C0018666666 [ 2268 ]
6663C00
1110 DATA 300C183C667E6600300C7E607 [ 2298 ]
C607E00
1120 DATA 300C7E1818187E00300C3C666 [ 2285 ]
6663C00
1130 DATA 300C666666663C006600663C1 [ 2244 ]
8181800
1140 DATA 6600183C667E660066007E607 [ 2281 ]
C607E00
1150 DATA 66007E1818187E0066003C666 [ 2268 ]
6663C00
1160 DATA 6600666666663C003C66C0C06 [ 2279 ]
63C0018
1170 DATA 3E78D8FED8D8DE001800183C6 [ 2382 ]
67E6600
1180 DATA 7ACCCED6E666BC00324C00667 [ 2380 ]
66E6600
1190 DATA 324C003C667E6600324C3C666 [ 2295 ]
6663C00
1200 DATA C0300C30CC30C000061860186 [ 2238 ]
6180600
1210 DATA 060C7E187E3060000000324C0 [ 2233 ]
07E0000
1220 DATA 0C30780C7CCC76000E1818701 [ 2297 ]
8180E00
1230 DATA 0C30003818183C000C30003C6 [ 2243 ]
6663C00
1240 DATA 0C3000666663E001866780C7 [ 2295 ]
CCC7600
1250 DATA 18663C667E603C00186600381 [ 2265 ]
8183C00
1260 DATA 1866003C66663C00186600666 [ 2251 ]
6663C00
1270 DATA 7CC6DEDEDEC07C007018180E1 [ 2374 ]
8187000
1280 DATA 300C003818183C00300C003C6 [ 2248 ]
6663C00
1290 DATA 181818181818180066006666 [ 2253 ]
63E067C
1300 DATA 6600780C7CCC760066003C667 [ 2303 ]
E603C00
1310 DATA 6600381818183C006600003C6 [ 2231 ]
6663C00
1320 DATA 660000666663E00C06030180 [ 2216 ]
C060200
1330 DATA 00006C1A7ED86E001800780C7 [ 2327 ]
CCC7600
1340 DATA 00007ACCD666BC00324C00DC6 [ 2322 ]
6666600
1350 DATA 324C780C7CCC7600000000FF0 [ 2298 ]
8080808
1360 DATA F8F8F0F0E0E0C0C01F1F0F0F0 [ 2378 ]
7070303
1370 DATA FFFEFCF8F0E0C080FF7F3F1F0 [ 2467 ]
F070301
```

DISCOLOGY

Version 5.0 pour Amstrad CPC

DISCOLOGY
Tapez |cpm pour démarrer
MERIDIEN INFORMATIQUE
5/7 la Canebière 13001 MARSEILLE tél 91 94 15 53

POUR VOUS SURPASSER

Que vous soyez un crack ou un débutant, DISCOLOGY vous propulse au-delà des limites du possible. Vous avez, en un clin d'œil, l'accès intégral à l'information contenue dans vos disquettes.

Son Désassembleur intelligent, son Listeur Basic et sa boîte à outils complète ouvrent pour vous les portes de l'inaccessible.

Pour toutes vos questions, l'Aide Intégrée apporte des réponses claires et intelligentes. Pour toutes vos ambitions, la Notice Technique vous livre les clés d'un monde inconnu.

Un Editeur ultra-puissant, un Copieur hyper-performant, un Exploreur qui n'a pas froid aux yeux : un cocktail détonant qui vous permet de vous surpasser. Avec la version 5.0, toutes les manipulations deviennent faciles, tous les horizons s'ouvrent devant vous. Alors, n'attendez plus ! Partez à la découverte de la dimension cachée de vos disquettes.

7 POINTS FORTS :

> la facilité : Fenêtres, Menus Déroulants, Aide Intégrée.
> La vitesse : 160 Ko de Langage Machine pur.
> La performance : la copie de sauvegarde intégrale pour vos disquettes et cassettes. Encore plus rapide, encore plus puissante.
> La précision : un manuel complet et une notice technique approfondie.
> L'inédit : un Editeur universel de secteurs, un Désassembleur Z 80, un Listeur Basic, un Exploreur en "Temps Réel"...
> La compatibilité : la gestion intégrale des extensions mémoire, des lecteurs 5 1/4 pouces.
> Les références : des milliers d'utilisateurs satisfaits en France comme à l'Etranger. DISCOLOGY est reconnu et acclamé par la Presse Internationale.

La disquette DISCOLOGY est disponible immédiatement chez votre revendeur.
Vous pouvez également la commander sans frais de port à : **MERIDIEN Informatique 5 et 7, La Canebière - 13001 Marseille - Tél. : 91.94.15.53**

- Master Save, copieur de disquettes, est disponible au prix de 190 F.
- Si vous commandez DISCOLOGY et possédez déjà Master Save, vous ne paierez que la différence.



AM.HS 5

BON DE COMMANDE

Version 5.0
Disponibilité immédiate.

☐ Je commande DISCOLOGY au prix de 350 F
☐ Je commande Master Save au prix de 190 F
☐ Je possède déjà Master Save et je commande DISCOLOGY. Je joins ma disquette Master Save et je ne paye que 160 F

Je règle ma commande :
☐ par chèque joint (port gratuit)
☐ contre-remboursement (+ 30 F de frais de port)

Nom : ______________________ Prénom : ______________________

Adresse : ____

Code Postal : ______________ Ville : ______________ Tél. : ______________

A retourner à MERIDIEN Informatique - 5 et 7, La Canebière - 13001 MARSEILLE

MERIDIEN INFORMATIQUE